3 Funktionale Werkzeuganforderungen

In den nachfolgenden Abschnitten, die nach der Grobstruktur einer SEU gegliedert sind, werden die einzelnen Leistungseinheiten, aus denen sich das SEU-Referenzmodell zusammensetzt, zusammen mit ihren Anforderungen beschrieben.

3.1 Hinweise zur Beschreibung der Werkzeuganforderungen

Jede funktionale Werkzeuganforderung besitzt eine alphanumerische Bezeichnung und wird hierdurch eindeutig identifizierbar.

Die Anforderungen sind hierarchisch geordnet, um die Anwendbarkeit und Verständlichkeit für unterschiedliche Nutzer sicherzustellen. Die so entstehenden Hierarchiebäume sind in ihrer Tiefe nicht festgelegt.

Die Bezeichnung einer Anforderung erfolgt nach folgendem Schema:

An.p.q

Es gilt:

A	=	LOBV	für Objektverwaltung
A	=	LNSS	für Nutzerschnittstelle
A	=	LSES	für Systementwicklung steuern
A	=	LSIC	für Sicherheit
A	=	LSE	für Systemerstellung
A	=	LQS	für Qualitätssicherung
A	=	LKM	für Konfigurationsmanagement
A	=	LPM	für Projektmanagement

„n“ ist die Nummer der jeweiligen Leistungseinheit

Es gilt:

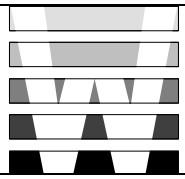
p	=	S	für Anforderungen an die Schnittstellen
p	=	M	für Anforderungen an die Methodenunterstützung
p	=	F	für Anforderungen an die Funktionen
p	=	So	für sonstige Anforderungen

„q“ bezeichnet eine Anforderung innerhalb einer Leistungseinheit und besteht aus Zahlen, die durch Punkte getrennt sind; die Anzahl der Zahlen gibt die Hierarchiestufe der Anforderung an.

Beispiele:

- LSE01.S.2.1 und LSE01.S.2.2 sind Verfeinerungen bzw. Präzisierungen der Anforderung LSE01.S.2 (Leistungseinheit „Anforderungen erfassen“, Anforderungen an die Schnittstellen).
- LKM01.So identifiziert die sonstigen Anforderungen der Leistungseinheit „Konfigurationsplanung und -kontrolle unterstützen“.

Ist eine Anforderung nicht ohne weiteres verständlich, so wird sie in einem folgenden Absatz erläutert. Erläuterungen werden in Kursivschrift von den Anforderungen abgesetzt.



3.2 Leistungskomplex „Objektverwaltung“

3.2.1 Leistungseinheit LOBV01 „Datenstrukturen und Daten der SEU erstellen, speichern, verwalten“

3.2.1.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.2.1.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur dauerhaften Aufbewahrung der Ergebnisse, Produkte und Objekte gestellt.

Diese Leistungseinheit enthält u. a. Anforderungen an ein Repository, über das Werkzeuge miteinander kommunizieren können.

Durch diese Leistungseinheit werden folgende Konzepte gefordert:

- Datenmodell mit den Fähigkeiten zur Definition von
  - komplexen Objekten und
  - langen Datenfeldern
- Navigation zu Objekten
- Schema- und View-Mechanismen
- Nutzersynchronisation
  - Transaktionen
  - lange Transaktionen

3.2.1.3 Anforderungen

3.2.1.3.1 Anforderungen an die Schnittstellen

LOBV01.S.1 Daten- und Nachrichtenschnittstellen
Es ist möglich, über offengelegte, dokumentierte Schnittstellen Daten und Nachrichten ohne weitere Transformation mit anderen Werkzeugen auszutauschen.

„Ohne weitere Transformation“ heißt, daß der Daten- und Nachrichtenaustausch zwischen einzelnen Werkzeugen über normierte Ausschnitte aus der Schemadefinition von LOBV01 (z. B. Daten zur Datenflußmodellierung über eine normierte Metastruktur zu Datenflußdiagrammen) erfolgt.

LOBV01.S.2 Schnittstelle zur Leistungseinheit LSIC03 „Zugriffskontrolle“
Alle Zugriffsversuche auf Objekte, die der Objektverwaltung unterliegen, werden zunächst der Zugriffskontrolle unterzogen.

3.2.1.3.2 Anforderungen an die Methodenunterstützung

keine

3.2.1.3.3 Anforderungen an die Funktionen

LOBV01.F.1 Zentrale Datenverwaltung

LOBV01.F.1.1 Repository
Es existiert eine logisch zentrale Datenverwaltung (Repository), in der die Beschreibung und Verwaltung aller Metadaten der SEU sowie deren Beziehungen zueinander vorgenommen werden.

Die Datenverwaltung muß nicht physisch zusammenhängend gespeichert sein.

LOBV01.F.1.2 Grafikspeicherung
Es ist möglich, zu Grafiken die ihnen zugrundeliegenden Informationen in einer zentralen Datenbasis zu speichern. Aus diesen Informationen können die Grafiken automatisch reproduziert werden.

Die den Grafiken zugrundeliegenden (Teil-) Informationen können mit anderen Informationen in der zentralen Datenbasis verknüpft werden.

LOBV01.F.1.3 Grafikänderungen
Änderungen in der Grafik werden automatisch in den ihnen zugrundegelegten Informationen in der zentralen Datenbasis nachgezogen und umgekehrt.

LOBV01.F.1.4 Vordefinierte Submodelle
Es gibt vordefinierte Metamodelle, in denen alle Metaobjekte einer minimalen SEU-Ausstattung beschrieben sind.

LOBV01.F.1.5 Verknüpfungen
Es ist möglich, alle Objekte miteinander zu verknüpfen und die Verknüpfungen wieder aufzulösen.

Verknüpfungen bzw. Auflösung der Verknüpfungen müssen vom Nutzer angestoßen werden können. So müssen sich z. B. bei der Datenflußmodellierung Datenspeichern, Datenflüssen und Prozessen eines bestimmten Datenflußdiagramms E/R-Teilmodelle durch den Nutzer zuordnen lassen. Diese Zuordnung kann vom Nutzer wieder aufgelöst werden.

LOBV01.F.2 Schemadefinition

LOBV01.F.2.1 Datenbeschreibungssprache
Es existiert eine Datenbeschreibungssprache zur Formulierung des Datenschemas einschließlich des Metadatenschemas.

LOBV01.F.2.2 Objekttypen
Es ist möglich, Objekttypen zu definieren.

LOBV01.F.2.3 Assoziation
Es ist möglich, assoziative Objekttypen zu definieren.

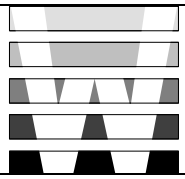
LOBV01.F.2.4 Subtypen
Es ist möglich, Subtypen zu definieren.

LOBV01.F.2.5 Binäre Beziehungstypen
Es ist möglich, binäre Beziehungstypen zwischen Objekttypen zu definieren.

LOBV01.F.2.6 Kardinalitäten
Es ist möglich, Kardinalitäten für Beziehungstypen anzugeben.

LOBV01.F.2.7 Attribute
Es ist möglich, Attribute für Objekttypen zu definieren.

LOBV01.F.2.8 Unstrukturierte Attribute
Es ist möglich, unstrukturierte Attribute beliebiger Länge (long fields) zu definieren.



LOBV01.F.2.9 Domänen
Es ist möglich, Domänen zu definieren und diese Objekten zuzuordnen.

LOBV01.F.2.10 Strukturschachtelung
Es ist möglich, komplexe Objekttypen zu strukturieren.

LOBV01.F.3 Datenbasen

LOBV01.F.3.1 Views
Es ist möglich, für eine Datenbasis Views zu definieren.

LOBV01.F.3.2 Nutzerzuordnung
Es ist möglich, Datenbasen Nutzern zuzuordnen.

LOBV01.F.3.3 Privat
Es ist möglich, öffentliche und private Datenbasen einzurichten.

LOBV01.F.3.4 Projektspezifisch
Es ist möglich, projektspezifische Datenbasen einzurichten.

LOBV01.F.4 Schlüssel

LOBV01.F.4.1 Primärschlüssel
Es ist möglich, für jeden Objekttyp einen Primärschlüssel zu definieren.

LOBV01.F.4.2 Sekundärschlüssel
Es ist möglich, Sekundärschlüssel zu definieren.

LOBV01.F.5 Zugriff
Es ist grundsätzlich möglich, alle Objekte zu lesen, zu schreiben, zu ändern und zu löschen.

LOBV01.F.6 Persistenz
Alle Objekte (einschließlich Metaobjekte) werden zwischen zwei Zugriffen dauerhaft aufbewahrt.

LOBV01.F.7 Schemaänderungen

LOBV01.F.7.1 Metaobjekte
Alle Metaobjekte des Schemas (Objekttypen, Beziehungstypen, Views, usw.) müssen gelesen, geschrieben, geändert und gelöscht werden können.

LOBV01.F.7.2 Schemaänderung
Es ist möglich, das Schema zu verändern.

LOBV01.F.7.3 Datenübernahme
Es ist möglich, bei Schemaänderungen den vorherigen Datenbestand in das neue Schema zu übertragen.

LOBV01.F.8 Attribute
Die Attribute aller Objekte müssen gelesen und bis auf Schlüsselattribute geändert werden können.

LOBV01.F.9 Mehrnutzerfähigkeit
Es ist möglich, daß alle Operationen gleichzeitig von mehreren Nutzern ausgeführt werden.

LOBV01.F.10 Transaktionen

LOBV01.F.10.1 Transaktionen
Es ist möglich, Transaktionen durchzuführen.

LOBV01.F.10.2 Rollback
Es ist möglich, beim Abbruch einer Transaktion automatisch auf den Zustand vor Transaktionsbeginn zurückzusetzen.

LOBV01.F.10.3 Erzwungener Rollback
Es ist möglich, eine Transaktion explizit zurückzusetzen (Rollback).

LOBV01.F.10.4 Schachtelung
Es ist möglich, Transaktionen zu schachteln.

LOBV01.F.11 Strukturierte Objekte (komplexe Objekte)

LOBV01.F.11.1 Strukturierte Objekte
Es ist möglich, strukturierte Objekte zu definieren und zu verwalten.

LOBV01.F.11.2 Einheit
Es existieren Operatoren zum Erzeugen, Bearbeiten und Löschen, die strukturierte Objekte als Einheit behandeln.

LOBV01.F.12 Betriebsarten

LOBV01.F.12.1 Online
Der Zugriff auf die Daten ist online möglich.

LOBV01.F.12.2 Batch
Der Zugriff auf die Daten ist im Hintergrund (Batch) möglich.

3.2.1.3.4 Sonstige Anforderungen

keine

3.2.2 Leistungseinheit LOBV02 „Datenstrukturen und Daten der SEU auswerten, abfragen“

3.2.2.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.2.2.2 Kurzcharakteristik

In dieser Leistungseinheit werden alle Anforderungen an Werkzeuge zusammengefaßt, die zur effektiven Auswertung der SEU-Daten erfüllt sein müssen, insbesondere unter dem Aspekt der passiven Abfrage und Auswertung der SEU-Daten. Die Auswertung kann dabei z. B. im Rahmen der Erstellung eines Dokuments („Report“) oder auch für die Erfüllung von Werkzeugleistungen geschehen, die dem Nutzer nicht unbedingt direkt zugänglich gemacht werden.

Alle mit (R) markierten Anforderungen sind nur auf relationale Datenbasen anwendbar.

3.2.2.3 Anforderungen

3.2.2.3.1 Anforderungen an die Schnittstellen

LOBV02.S.1 Schnittstelle zur Leistungseinheit LSIC03 „Zugriffskontrolle“
Alle Zugriffsversuche auf Objekte, die der Objektverwaltung unterliegen, werden zunächst der Zugriffskontrolle unterzogen.

3.2.2.3.2 Anforderungen an die Methodenunterstützung

keine

3.2.2.3.3 Anforderungen an die Funktionen

LOBV02.F.1 Auswerteoperatoren (R)
Zur Auswertung und Abfrage von SEU-Daten stehen Selektion, Projektion und Join als Auswerteoperatoren zur Verfügung.

LOBV02.F.2 Interaktion
Es ist möglich, Abfragen interaktiv durchzuführen.

LOBV02.F.3 Auswahlkriterien für alle Auswerteoperatoren (R)

LOBV02.F.3.1 Vergleich
Es ist möglich, Objekte unter Verwendung der Operatoren „gleich“, „ungleich“, „kleiner“ bzw. „größer“ anhand ihrer Attribute auszuwählen.

LOBV02.F.3.2 Berechnung
Es ist möglich, das Auswahlkriterium bei numerischen Attributen unter Verwendung von Operatoren zur Addition, Subtraktion, Multiplikation und Division zu formulieren.

LOBV02.F.3.3 Konstanten
Es ist möglich, für die Selektion Attribute nicht nur untereinander, sondern auch mit Konstanten der jeweiligen Domäne zu vergleichen.

LOBV02.F.3.4 Boolesche Operatoren
Es ist möglich, die Selektion mittels der Booleschen Operatoren „und“, „oder“ und „nicht“ und der Klammerung zu formulieren.

LOBV02.F.3.5 Schlüssel
Es ist möglich, die Auswahlkriterien mit Schlüsselattributen zu formulieren.

LOBV02.F.4 Schachtelung (R)
Es ist möglich, Selektion, Join und Projektion zu schachteln.

LOBV02.F.5 Verkettung (R)
Es ist möglich, die Ergebnisse von Selektion, Join oder Projektion als Ausgangspunkt weiterer Abfragen zu verwenden.

LOBV02.F.6 Strukturierte Objekte
Strukturierte Objekte werden bei allen Abfragen als Einheit betrachtet.

LOBV02.F.7 Schema
Es ist möglich, in der gleichen Art wie für Objekte (Instanzen) alle Abfragen auch für Schemainformationen (Metaobjekte) auszuführen.

LOBV02.F.8 Leistungsdaten
Es ist möglich, Leistungsdaten (statistische Informationen, Durchsatz, usw.) abzufragen.

LOBV02.F.9 Reports

LOBV02.F.9.1 Liste
Es ist möglich, Abfrageergebnisse als Liste aufzubereiten.

LOBV02.F.9.2 Gruppenwechsel
Es ist möglich, innerhalb eines Reports Gruppenwechsel zu definieren, und daran besondere Aktionen (mindestens Summenbildung, Seitenwechsel und Zählerschaltungen) zu knüpfen.

LOBV02.F.9.3 Reporttext
Es ist möglich, Reports mit erläuterndem Text (z. B. Überschrift, Spaltenerklärungen, usw.) zu versehen.

LOBV02.F.9.4 Druck
Es ist möglich, Reports direkt zu drucken oder als druckfertige Datei aufzubewahren.

LOBV02.F.10 Prozeduren

LOBV02.F.10.1 Folge
Es ist möglich, eine Folge von Auswerteoperationen zu einer Prozedur zusammenzufassen.

LOBV02.F.10.2 Parameter
Es ist möglich, Prozeduren mit Parametern zu versehen.

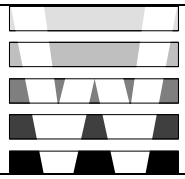
LOBV02.F.10.3 Aufruf
Es ist möglich, Abfragen mit Prozeduraufrufen zu formulieren.

LOBV02.F.10.4 Schachtelung
Es ist möglich, daß Prozeduren andere Prozeduren aufrufen.

LOBV02.F.10.5 Prozedursprache
Es ist möglich, Datenstrukturen und Daten der SEU, die durch LOBV01 verwaltet werden, mit einer Prozedursprache auszuwerten und abzufragen.

LOBV02.F.11 Cursor

LOBV02.F.11.1 Erzeugen
Es ist möglich, den Cursor zu öffnen und zu schließen.



LOBV02.F.11.2 Verändern
Es ist möglich, die Elemente eines Cursors zu lesen und zu verändern.

LOBV02.F.11.3 Aktive Cursor
Es ist möglich, daß gleichzeitig mehrere Cursor offen sind.

LOBV02.F.12 Views
Es ist möglich, alle Abfragen auf Views zu beschränken.

3.2.2.3.4 Sonstige Anforderungen

keine

3.2.3 Leistungseinheit LOBV03 „Datenstrukturen und Daten der SEU exportieren, importieren“

3.2.3.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.2.3.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zusammengefaßt, die den Datenaustausch zwischen (verschiedenen) SEU ermöglichen. Beim Export wird ein Ausschnitt der Objekte und Beziehungen der Datenbasis in eine Datenbeschreibungssprache übertragen und auf ein geeignetes Medium (Banddatei, Leitung, usw.) ausgegeben. Beim Import wird die Darstellung als Datenbeschreibungssprache eingelesen und in Form von Objekten und Beziehungen Teil der Datenbasis.

Damit der Datenaustausch zwischen unterschiedlichen SEU möglich ist, müssen die Datenstrukturen der unterschiedlichen SEU auf ein miteinander verträgliches Format (kompatibles Schema) gebracht werden, wobei zusätzlich zu prüfen ist, ob die Semantik der auszutauschenden Datenstrukturen gleich ist.

Jede Datenkommunikation mit SEU-externen Stellen wird über LOBV03 abgewickelt. Im folgenden wird die exportierende SEU als „Datenquelle“ und die importierende als „Datensenke“ bezeichnet.

3.2.3.3 Anforderungen

3.2.3.3.1 Anforderungen an die Schnittstellen

LOBV03.S.1 Schnittstelle zur Leistungseinheit LSIC03 „Zugriffskontrolle“
Alle Zugriffsversuche auf Objekte, die der Objektverwaltung unterliegen, werden zunächst der Zugriffskontrolle unterzogen.

3.2.3.3.2 Anforderungen an die Methodenunterstützung

keine

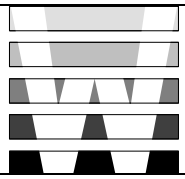
3.2.3.3.3 Anforderungen an die Funktionen

LOBV03.F.1 Dateitransfer
Alle exportierten Informationen werden als eine oder mehrere Dateien gespeichert, die im Dateisystem verwaltet werden können.

LOBV03.F.2 Stückelung
Es ist möglich, beim Export über die Aufteilung in eine oder mehrere Einzeldateien zu entscheiden.

LOBV03.F.3 Datenträger
Es ist möglich, alle exportierten Informationen auf externen Medien (Band, Kassette, Platte) zu speichern.

LOBV03.F.4 Neutraler Export
Der Export von Informationen hat keinen Einfluß auf den Inhalt der Datenquelle.



LOBV03.F.5 Invers
Ein Export aus der Datenquelle mit nachfolgendem Import zurück in die Datenquelle hat keine Änderung an Inhalt und Zustand der Datenquelle zur Folge.

LOBV03.F.6 Replace/Insert
Es ist möglich, beim Import zu wählen zwischen Überschreiben von Informationen, die bereits in der Datensenke vorhanden sind (replace), und dem Anlegen einer neuen Version in der Datensenke (insert).

LOBV03.F.7 Transaktion
Export und Import werden als Transaktionen durchgeführt.

Das heißt, daß beide Vorgänge unterbrechbar sind und der Vorzustand ohne Einschränkungen wiederhergestellt werden kann.

3.2.3.3.4 Sonstige Anforderungen

keine

3.3 Leistungskomplex „Nutzerschnittstelle“

3.3.1 Leistungseinheit LNSS01 „Funktionen der Nutzeroberfläche“

3.3.1.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.3.1.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an die Nutzeroberfläche der im Rahmen einer SEU eingesetzten Werkzeuge definiert. Die Nutzeroberfläche legt das Erscheinungsbild und die Bedienung des Werkzeugs fest. Ziel der Leistungseinheit ist es, daß sich alle Werkzeuge der SEU dem Nutzer durchgängig mit einer einheitlichen Nutzeroberfläche präsentieren. Der Umgang mit der Nutzeroberfläche sollte für den Nutzer intuitiv und leicht zu erlernen sein.

3.3.1.3 Anforderungen

3.3.1.3.1 Anforderungen an die Schnittstellen

LNSS01.S.1 Schnittstelle zur Leistungseinheit LSIC02 „Identifikation und Authentisierung“
Es ist sichergestellt, daß Subjekte einer Zugangskontrolle über LSIC02 unterzogen wurden, bevor sie weitere Aktionen über die Nutzerschnittstelle tätigen.

3.3.1.3.2 Anforderungen an die Methodenunterstützung

keine

3.3.1.3.3 Anforderungen an die Funktionen

LNSS01.F.1 Bedienerführung

LNSS01.F.1.1 Interaktive Bedienerführung
Es existiert eine interaktive Bedienerführung.

LNSS01.F.1.2 Stufen der Bedienerführung
Es ist möglich, zwischen verschiedenen Stufen der Bedienerführung zu wählen.
Mindestens kann zwischen Erstnutzer- und Expertenmodus gewählt werden.

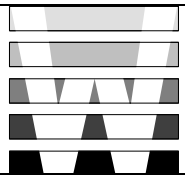
LNSS01.F.1.3 Eingabemedien

LNSS01.F.1.3.1 Zeigeinstrument
Die Auswahl von Objekten und Operationen erfolgt durch ein Zeigeinstrument.
Das Zeigeinstrument ist normalerweise eine Maus.

LNSS01.F.1.3.2 Tastatureingabe
Es ist möglich, die Nutzeroberfläche mit Hilfe der Tastatur zu steuern.

LNSS01.F.1.3.3 Funktionstasten
Es ist möglich, die Nutzeroberfläche über Funktionstasten zu steuern.

LNSS01.F.1.3.4 Funktionstastenbelegung
Es ist möglich, die Belegung der Funktionstasten zu ändern.



LNSS01.F.1.3.5 Makroabspeicherung
Es ist möglich, Tastenfolgen als Makro unter einem Namen abzuspeichern.

LNSS01.F.1.3.6 Makroaufruf
Es ist möglich, als Makro abgespeicherte Tastenfolgen aufzurufen.

LNSS01.F.1.4 „Undo“-Operation
Es ist möglich, die letzte Operation aufzuheben, falls es sich nicht um eine extra zu bestätigende Operation handelt.

LNSS01.F.1.5 Operationsabbruch
Es ist möglich, jede Operation abzubrechen. Es erscheint dann eine entsprechende Meldung.

Dies gilt auch für Operationen, die im Hintergrund ablaufen (z. B. Prüfungen).

LNSS01.F.2 Fenstertechnik

LNSS01.F.2.1 Dialogabwicklung via Fenster
Der Dialog mit dem Nutzer wird über Fenstertechnik abgewickelt.

Unterstützt die Hardware keinen Grafikmodus, so sollten die Fenster durch Trennlinien visualisiert werden.

LNSS01.F.2.2 Manipulation von Fenstern
Es ist möglich, Fenster zu öffnen, zu schließen und zu verschieben.

Fenster können auch über die Begrenzung der physischen Bildschirmgröße hinaus verschoben werden.

LNSS01.F.2.3 Ikonen
Es ist möglich, Fenster in Ikonen (Piktogramme) umzuwandeln.

LNSS01.F.2.4 Überlappende Fenster
Es ist möglich, mehrere Fenster gleichzeitig, auch überlappend, am Bildschirm anzuzeigen.

LNSS01.F.2.5 Vorne/hinten
Es ist möglich, Fenster nach vorne oder nach hinten zu bringen.

LNSS01.F.2.6 Ändern der Fenstergröße

LNSS01.F.2.6.1 Beliebige Fenstergröße
Es ist möglich, die Größe eines Fensters frei skalierbar festzulegen.

LNSS01.F.2.6.2 Bildschirmgröße
Es ist möglich, ein Fenster auf Bildschirmgröße zu vergrößern.

LNSS01.F.2.7 Verschiebebalken
Es ist möglich, mit Hilfe von Verschiebebalken über die Gesamtinformation zu navigieren. Der sichtbare Teil der Gesamtinformation wird im Fenster angezeigt.

LNSS01.F.2.8 Vergrößerte/Verkleinerte Darstellung
Es ist möglich, eine vergrößerte oder verkleinerte Darstellung von Fensterinhalten zu wählen (Zoom).

LNSS01.F.2.9 Überblicksfenster
Es ist möglich, den Fensterinhalt in einem eigenen Fenster überblicksartig darzustellen.

LNSS01.F.2.10 Statuszeile
Es ist möglich, zu einem Fenster eine Statuszeile einzublenden, die statistische Informationen anzeigt.

Dazu zählen u. a. Informationen, wieviele Objekte zum Fensterinhalt gehören, wieviele von diesen aktuell markiert oder ausgeblendet sind.

LNSS01.F.2.11 Raster
Es ist möglich, zu einem Fenster ein Raster einzublenden, das die Positionierung von Objekten vereinfacht.

LNSS01.F.2.12 Lineal
Es ist möglich, zu einem Fenster ein Lineal einzublenden.

LNSS01.F.3 Aktualisierung

LNSS01.F.3.1 Bildschirm
Es ist möglich, den gesamten Bildschirm wiederaufzufrischen, um die Anzeige auf den neuesten Stand zu bringen.

LNSS01.F.3.2 Fenster
Es ist möglich, ein bestimmtes Fenster wiederaufzufrischen, um die Anzeige auf den neuesten Stand zu bringen.

LNSS01.F.4 Meldungen

LNSS01.F.4.1 Typen von Meldungen
Es gibt mindestens eine Unterscheidung zwischen Warnungen und Fehlern.

LNSS01.F.4.2 Fenster für Meldungen
Meldungen erscheinen in einem getrennten Fenster oder optisch abgetrennten Bildschirmbereich.

LNSS01.F.4.3 Einheitliche Form der Meldungen
Meldungen sind einheitlich gestaltet und formuliert.

LNSS01.F.4.4 Ausgabe von Meldungen
Es ist möglich, Meldungen auszudrucken oder in eine Datei umzuleiten.

LNSS01.F.4.5 Sprachliche Fassung
Es ist möglich, die Meldungen sprachlich umgebungskonform zu verfassen.

LNSS01.F.5 Betriebssystemkommandos
Es ist möglich, Kommandos an das Betriebssystem abzusetzen.

LNSS01.F.6 Rückkehr zum Betriebssystem
Eine Rückkehr zum Betriebssystem muß explizit bestätigt werden.

LNSS01.F.7 Ausgabe des Bildschirminhaltes
Es ist möglich, den aktuellen Bildschirminhalt auszudrucken oder in eine Datei umzuleiten.

3.3.1.3.4 Sonstige Anforderungen

keine

3.3.2 Leistungseinheit LNSS02 „Nutzerprofil festlegen“

3.3.2.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.3.2.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge hinsichtlich der für einzelne Personen oder Personengruppen möglichen Gestaltungsvarianten der Nutzerprofile gestellt. Es kann damit z. B. innerhalb der festgelegten Richtlinien zur Nutzeroberfläche gewählt werden, welche Varianten der Nutzeroberflächengestaltung verwendet werden soll. Dies kann z. B. die Gestaltung spezieller Bildschirmlayouts betreffen.

3.3.2.3 Anforderungen

3.3.2.3.1 Anforderungen an die Schnittstellen

keine

3.3.2.3.2 Anforderungen an die Methodenunterstützung

keine

3.3.2.3.3 Anforderungen an die Funktionen

LNSS02.F.1 Symbole für methodengebundene Objekte
Es ist möglich, für die Darstellung von methodengebundenen Objekten zwischen mehreren Symbolen auszuwählen.

Eine Funktion in einem Hierarchie-Diagramm kann z. B. durch ein Rechteck, einen Kreis oder eine Ellipse repräsentiert werden.

LNSS02.F.2 Voreinstellungen für Fenster

LNSS02.F.2.1 Fenstergröße
Es ist möglich festzulegen, ob ein Fenster nach Öffnen in normaler, minimierter oder maximierter Größe auf dem Bildschirm erscheinen soll.

Normale Größe ist die Voreinstellung für Fenster vom Betriebssystem her, bei minimierter Größe erscheint nur eine Ikone (Piktogramm), die das Fenster repräsentiert, bei maximierter Größe wird der Bildschirm vollständig vom betreffenden Fenster ausgefüllt.

LNSS02.F.2.2 Fensterattribute
Es ist möglich festzulegen, ob einem Fenster ein Raster, ein Lineal und/oder eine Statuszeile zugeordnet werden soll.

LNSS02.F.2.3 Überblicksfenster
Es ist möglich festzulegen, ob zu einem Fenster prinzipiell ein Überblicksfenster zur schnelleren Positionierung während der Bearbeitung der Fensterinhalte zugeordnet werden soll.

LNSS02.F.3 Voreinstellungen für Grafiken

LNSS02.F.3.1 Darstellung von Grafiken
Es ist möglich festzulegen, ob eine Grafik prinzipiell in Originalgröße oder angepaßt an die jeweils verfügbare Fenstergröße dargestellt werden soll.

LNSS02.F.3.2 Skalierung von Objekten
Es ist möglich, für die grafische Darstellung von Objekten eine standardmäßige Skalierung der entsprechenden Symbole voreinzustellen.

LNSS02.F.3.3 Präsentationsqualität
Es ist möglich festzulegen, ob eine Grafik prinzipiell in Präsentations- oder in Entwurfsqualität dargestellt werden soll.

LNSS02.F.4 Voreinstellungen für Text

LNSS02.F.4.1 Zeichensatz
Es ist möglich, für Texte den prinzipiell zu verwendenden Zeichensatz festzulegen.

LNSS02.F.4.2 Schriftgröße
Es ist möglich, für Texte die prinzipiell zu verwendende Schriftgröße festzulegen.

LNSS02.F.5 Voreinstellungen für Dokumente

LNSS02.F.5.1 Kopfzeilen, Fußzeilen
Es ist möglich, Inhalte und Positionierung von Kopf- oder Fußzeilen für den Ausdruck von Dokumenten festzulegen.

LNSS02.F.5.2 Dateinamen
Es ist möglich, für den Namen von Dateien, in denen Dokumente gespeichert werden sollen, einen Standardwert festzulegen.

LNSS02.F.6 Voreinstellungen von Farben

LNSS02.F.6.1 Vorder-/Hintergrundfarben
Es ist möglich, in einem Fenster Farben für Vorder- und Hintergrund festzulegen.

LNSS02.F.6.2 Text
Es ist möglich, die Farbe eines Textes festzulegen.

LNSS02.F.6.3 Markierter Text
Es ist möglich, die Farbe eines markierten Textes festzulegen.

LNSS02.F.6.4 Grafisches Symbol
Es ist möglich, die Farbe eines grafischen Symbols festzulegen.

LNSS02.F.6.5 Markiertes grafisches Symbol
Es ist möglich, die Farbe eines markierten grafischen Symbols festzulegen.

LNSS02.F.6.6 Meldungen
Es ist möglich, die Farbe von Meldungen festzulegen.

LNSS02.F.6.7 Farbpalette
Es ist möglich, bei der Festlegung einer Farbe aus einer Farbpalette mit vordefinierten Farben zu wählen.

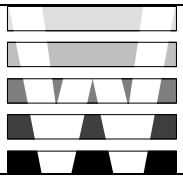
LNSS02.F.7 Voreinstellung für die Implementierungsumgebung

LNSS02.F.7.1 Implementierungssprache
Es ist möglich, die Programmiersprache, mit der das Zielsystem implementiert werden soll, festzulegen.

LNSS02.F.7.2 TP-Monitor
Es ist möglich, den für das Zielsystem zu verwendenden TP-Monitor festzulegen.

LNSS02.F.7.3 Datenbank-Managementsystem (DBMS)
Es ist möglich, das für das Zielsystem zu verwendende DBMS festzulegen.

3.3.2.3.4 Sonstige Anforderungen



keine

3.3.3 Leistungseinheit LNSS03 „Hilfefunktionen“

3.3.3.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.3.3.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge bezüglich der Hilfefunktionen im Dialogbetrieb (online) gestellt. Weiterhin werden in dieser Leistungseinheit Anforderungen an Werkzeuge bezüglich deren Dokumentation gestellt.

3.3.3.3 Anforderungen

3.3.3.3.1 Anforderungen an die Schnittstellen

keine

3.3.3.3.2 Anforderungen an die Methodenunterstützung

keine

3.3.3.3.3 Anforderungen an die Funktionen

LNSS03.F.1 Hilfefunktion

LNSS03.F.1.1 Hilfetextauswahl
Es ist möglich, zu Funktionen des Werkzeugs Hilfetexte anzufordern.

LNSS03.F.1.2 Hilfe zu Hilfe
Es ist möglich, einen Hilfetext zum Aufruf und zur Verwendung der Hilfefunktion anzufordern.

LNSS03.F.1.3 Tastaturbelegung
Es ist möglich, die aktuelle Tastaturbelegung anzuzeigen. Dies schließt die Belegung der Funktionstasten und Tastenkombinationen (z. B. Control, Esc) mit ein.

LNSS03.F.1.4 Kontextsensitiv
Die Hilfefunktion ist kontextsensitiv.

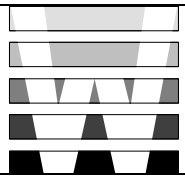
Der angebotene Hilfetext ist abhängig von der gerade bearbeiteten Operation und von der Position in der Anwendung.

LNSS03.F.1.5 Weitere Hilfe
Es ist möglich, zu bestimmten Stichworten innerhalb des Hilfetextes weitere Hilfe anzufordern.

Diese Stichworte sind besonders markiert, z. B. farbig unterlegt.

LNSS03.F.1.6 Letzte Fehlermeldung
Es ist möglich, Hilfe zur letzten Fehlermeldung anzufordern.

LNSS03.F.1.7 Hilfe über Index
Es ist möglich, Hilfe über eine alphabetisch sortierte Liste von Stichworten anzufordern.



LNSS03.F.1.8 Hilfegrad
Es ist möglich, den Grad der Hilfeunterstützung während der Sitzung zu verändern.

Beispielsweise kann zwischen den Graden „kurz“, „mittel“ und „lang“ ausgewählt werden. Ein Erstnutzer des Werkzeugs wird die „lange“ Hilfeunterstützung wählen, ein Fortgeschrittener die „kurze“.

LNSS03.F.1.9 Eigene Hilfetexte
Es ist möglich, selbst erstellte Hilfetexte in das Werkzeug zu integrieren.

Dies ist zum Beispiel sinnvoll, wenn auch selbst erstellte Funktionen in das Werkzeug integriert werden können.

LNSS03.F.2 Anzeige der Hilfe

LNSS03.F.2.1 Umschaltung
Es ist möglich, zwischen dem Hilfefenster und der Anwendung umzuschalten.

LNSS03.F.2.2 Gleichzeitige Darstellbarkeit
Hilfefenster und Anwendung sind gleichzeitig darstellbar.

LNSS03.F.2.3 Mehrere Hilfefenster
Mehrere Hilfefenster sind gleichzeitig darstellbar.

LNSS03.F.3 Drucken der Hilfe
Es ist möglich, den Hilfetext auszudrucken.

LNSS03.F.4 Dokumentation

LNSS03.F.4.1 Inhaltsverzeichnis
Die Dokumentation enthält ein Inhaltsverzeichnis.

LNSS03.F.4.2 Installationsanleitung
Die Dokumentation enthält eine Anleitung zur Installation.

LNSS03.F.4.3 Einführung
Die Dokumentation enthält eine Einführung für Erstnutzer.

LNSS03.F.4.4 Kurzreferenz
Die Dokumentation enthält eine Kurzreferenz.

In ihr werden überblicksartig die wichtigsten Funktionen mit ihren Parametern aufgelistet. Sie sollte möglichst handlich sein (z. B. als Faltblatt) und neben die Tastatur gelegt werden können, ohne die Arbeit zu behindern.

LNSS03.F.4.5 Funktionalität
Die Dokumentation beschreibt die Funktionalität des Werkzeugs vollständig.

Es sollten ausführliche Beispiele zu den einzelnen Funktionen vorhanden sein.

LNSS03.F.4.6 Meldungen
Die Dokumentation enthält eine vollständige Liste der Meldungen, ihrer möglichen Fehlerursachen und empfohlenen Abhilfemaßnahmen.

Diese Liste sollte nach Meldungsarten (Warnungen, Fehler, usw.) und innerhalb dieser Kategorien alphabetisch oder nach Fehlerschlüssel sortiert sein.

LNSS03.F.4.7 Besonderheiten
Die Dokumentation enthält eine Auflistung der implementierungsbedingten Besonderheiten.

Dazu gehören auch durch die Implementierung vorgegebene Einschränkungen (z. B. maximale Anzahl von Zeilen einer Datei, die mit einem Editor bearbeitet werden kann).

LNSS03.F.4.8 Kompatibilität
Die Dokumentation enthält eine Angabe über die Aufwärts- und Abwärtskompatibilität des Werkzeugs zu anderen Versionen dieses Werkzeugs.

LNSS03.F.4.9 Fehler
Die Dokumentation enthält eine Liste der bekannten Fehler.

LNSS03.F.4.10 Stichwortverzeichnis
Die Dokumentation enthält ein globales Stichwortverzeichnis.

Bei umfangreicher Dokumentation mit mehreren Bänden verweist das globale Stichwortverzeichnis auf die Seite im betroffenen Band. Dies erleichtert den gezielten Zugriff wesentlich.

LNSS03.F.4.11 Passend zum Werkzeug
Die Dokumentation bezieht sich auf die richtige Version des dokumentierten Werkzeugs.

3.3.3.3.4 Sonstige Anforderungen

LNSS03.So.1 Verständlichkeit
Die Hilfetexte sind klar verständlich und nachvollziehbar.

LNSS03.So.2 Sprachliche Fassung
Es ist möglich, die Meldungen sprachlich umgebungskonform zu verfassen.

LNSS03.So.3 Beispiele
Die Hilfetexte enthalten Beispiele.

3.3.4 Leistungseinheit LNSS04 „Dokumentation aufbereiten“

3.3.4.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.3.4.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Aufbereitung von Ergebnissen (Text und Grafik) im Sinne einer einheitlichen Dokumentation mit geeignetem Layout gestellt.

Während der Entwicklungsarbeit kann auch eine noch unvollständige Dokumentation erzeugt werden. Weiterhin besteht die Möglichkeit, durch gezielte Auswahl eine Teildokumentation zu erzeugen (unterschiedlicher Detaillierungsgrad, Interessensschwerpunkte, usw.).

Diese Leistungseinheit eignet sich nicht nur für die Entwicklungsdokumentation, sondern auch für die Erzeugung geeigneter Handbücher, die an den Endnutzer weitergegeben werden können. Außerdem kann sie für die Erstellung anderer Dokumente (z. B. Sachstandsberichte) genutzt werden.

3.3.4.3 Anforderungen

3.3.4.3.1 Anforderungen an die Schnittstellen

LNSS04.S.1 Eingangsschnittstelle zu LNSS05 „Texte editieren“
Es ist möglich, mit LNSS05 editierte Texte ohne weitere Transformation aus der Objektverwaltung zu übernehmen.

Ohne weitere Transformation bedeutet hier: Umlaute und Tabulatoren werden übernommen oder automatisch umgesetzt, Formatierungsinformationen bleiben erhalten.

LNSS04.S.2 Eingangsschnittstelle zu LNSS06 „Grafiken editieren“
Es ist möglich, mit LNSS06 editierte Grafiken ohne weitere Transformation aus der Objektverwaltung zu übernehmen.

3.3.4.3.2 Anforderungen an die Methodenunterstützung

keine

3.3.4.3.3 Anforderungen an die Funktionen

LNSS04.F.1 Formatierung

LNSS04.F.1.1 Zeichenformatierung

LNSS04.F.1.1.1 Fett/Kursiv/Unterstrichen
Es ist möglich, Text mindestens mit den Druckeigenschaften „fett“, „kursiv“ und „unterstrichen“ (auch kombiniert) zu kennzeichnen.

LNSS04.F.1.1.2 Verborgener Text
Es ist möglich, Text als „verborgen“ zu markieren.

Bei verborgenem bzw. unsichtbarem Text kann zum Zeitpunkt des Ausdruckes entschieden werden, ob dieser Text mit gedruckt wird oder nicht. Er eignet sich zur Berücksichtigung von Bearbeitungshinweisen, Anmerkungen, Kommentaren, usw.

LNSS04.F.1.2 Absatzformatierung

LNSS04.F.1.2.1 Links/Rechts/Zentriert/Block
Es ist möglich, Text linksbündig, rechtsbündig, zentriert und als Block auszurichten.

LNSS04.F.1.2.2 Zeilenbreite
Es ist möglich, die Breite einer Zeile festzulegen.

LNSS04.F.1.2.3 Einzug zweite Zeile
Es ist möglich, für die zweite und alle folgenden Zeilen eines Absatzes eine alternative Zeilenbreite vorzugeben.

Hierdurch lassen sich Aufzählungen (Strichaufzählung bzw. 1., 2., 3., ...) einfach eingeben.

LNSS04.F.1.2.4 Zusammenhängende Absätze

LNSS04.F.1.2.4.1 Absatztext
Es ist möglich zu bestimmen, daß alle Zeilen eines Absatzes nur zusammenhängend auf der gleichen Seite ausgedruckt werden.

Hierdurch kann der Aufwand für den Seitenumbruch erheblich reduziert werden.

LNSS04.F.1.2.4.2 Folgeabsätze
Es ist möglich zu bestimmen, daß der nachfolgende Absatz auf der gleichen Seite ausgedruckt wird.

LNSS04.F.1.3 Dokumentenformatierung

LNSS04.F.1.3.1 Druckseitengröße
Es ist möglich, die Seitengröße des Dokumentes zu bestimmen.

LNSS04.F.1.3.2 DIN A4
Es wird DIN A4 im Hochformat unterstützt

LNSS04.F.1.3.3 Layoutmuster

LNSS04.F.1.3.3.1 Mehrfachverwendung
Es ist möglich, zur einheitlichen Layoutgestaltung die Beschreibung von Zeichen- und Absatzformatierungen getrennt vom eigentlichen Dokument vorzunehmen und bei der Erstellung einer Dokumentation auf die vordefinierten Layoutmuster Bezug zu nehmen.

LNSS04.F.1.3.3.2 Beschreibungsumfang
Layoutmuster umfassen mindestens Zeichen– und Absatzformatierungen.

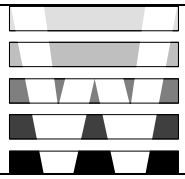
Zeichenformatierungen beinhalten Schriftart (fett/kursiv/unterstrichen), Schriftgröße und Schrifttyp (Font); Absatzformatierungen beinhalten Anfangs-, Ende- und Zeilenabstand, Absatzeinzug des gesamten Absatzes und der ersten Zeile und die Ausrichtung (links-, rechtsbündig, zentriert, Blocksatz).

LNSS04.F.1.4 Nachträgliche Formatänderung

LNSS04.F.1.4.1 Einzeländerung
Es ist möglich, eine vorgenommene Formatierung nachträglich zu ändern.

Dies betrifft die Zeichen-, Absatz- und Dokumentenformatierung.

LNSS04.F.1.4.2 Bereich
Es ist möglich, eine nachträgliche Änderung der Formatierung für mehrere zusammenhängende Zeichen oder Absätze gleichzeitig durchzuführen.



LNSS04.F.2 Seitenbearbeitung

LNSS04.F.2.1 Seitenumbruch

LNSS04.F.2.1.1 Automatisch
Automatischer Seitenumbruch ist möglich.

LNSS04.F.2.1.2 Erzwungen
Es ist möglich, die Position für einen Seitenumbruch vorzugeben.

LNSS04.F.2.2 Kopf-/Fußzeilen

LNSS04.F.2.2.1 Frei gestaltbar
Kopf- und Fußzeilen sind frei gestaltbar.

LNSS04.F.2.2.2 Mehrzeilig
Es ist möglich, daß Kopf- und Fußzeilen mehrzeilig sind.

LNSS04.F.2.2.3 Kapitelweise änderbar
Es ist möglich, den Inhalt von Kopf- und Fußzeilen kapitelweise zu ändern.

LNSS04.F.2.2.4 Grafikeinbindung
Es ist möglich, daß Kopf- und Fußzeilen Grafiken enthalten.

LNSS04.F.2.3 Seitennumerierung

LNSS04.F.2.3.1 Automatisch
Eine automatische Seitennumerierung mit arabischen Ziffern (1, 2, 3, ...) ist möglich.
Weitere mögliche Seitennumerierungen sind: römisch (groß oder klein), alphabetisch.

LNSS04.F.2.3.2 In Kopf- oder Fußzeilen
Es ist möglich, die Seitennumerierung in Kopf- oder Fußzeilen wahlweise zu integrieren.

LNSS04.F.2.3.3 Text vor/nach Seitenzahl
Es ist möglich, der Seitenzahl variablen Text/Buchstaben, usw. voran- oder nachzustellen.

LNSS04.F.2.3.4 Beginn beliebig
Es ist möglich, die Zahl, mit der die Seitennumerierung beginnt, vom Nutzer vorzugeben.

LNSS04.F.2.4 Seitenvorschau

LNSS04.F.2.4.1 Seitenumbruch erkenntlich
Im Bearbeitungsmodus ist erkennbar, an welcher Stelle ein Zeilen- oder Seitenumbruch stattfindet.

LNSS04.F.2.4.2 WYSIWYG
Eine Bildschirmansicht, die genau dem Ausdruck eines Dokumentes entspricht, ist möglich.
WYSIWYG bedeutet „What you see is what you get“.

LNSS04.F.3 Gliederung

LNSS04.F.3.1 Gliederung vorab
Es ist möglich, zunächst eine Gliederung des Dokumentes zu erstellen, in die nachträglich Texte/Grafiken eingefügt werden können.

LNSS04.F.3.2 Gliederung nachträglich
Es ist möglich, für ein Dokument nachträglich eine Gliederung zu erstellen.

LNSS04.F.3.3 Numerierung
Es ist möglich, eine Gliederung automatisch zu numerieren.

LNSS04.F.3.4 Arabische Ziffern
Die Numerierung der Gliederung ist mindestens mit durch Punkt getrennten arabischen Ziffern (z. B. 1.2.1.4) möglich.

LNSS04.F.3.5 Wechsel der Ansicht
Es ist möglich, zwischen einer Gliederungsansicht und einer Ansicht des Dokumentes zu wechseln.

Die Gliederungsansicht zeigt nur Gliederungspunkte und ermöglicht einen besseren Überblick über die Struktur eines Dokumentes.

LNSS04.F.3.6 Gliederungspunkte verschiebbar
Beim Verschieben eines Gliederungspunktes werden zugehörige Texte/Grafiken mit verschoben.

LNSS04.F.3.7 Einordnung änderbar
„Höherstufen“, „Herabstufen“, „als Text zurückstufen“ von Gliederungspunkten ist möglich.

LNSS04.F.4 Inhaltsverzeichnis

LNSS04.F.4.1 Automatische Erstellung
Es ist möglich, ein Inhaltsverzeichnis mit Seitenzahlangaben automatisch über das Gesamtdokument zu erstellen.

LNSS04.F.4.2 Berücksichtigung Gliederung
Es ist möglich, bei der Erstellung eines Inhaltsverzeichnisses die Informationen der „Gliederung“ zu berücksichtigen.

LNSS04.F.5 Index
Es ist möglich, einen Index mit Seitenzahlangaben automatisch über das Gesamtdokument zu erstellen.

LNSS04.F.6 Fußnoten
Es ist möglich, Fußnoten zu erstellen.

LNSS04.F.7 Kennzeichnung von Abbildungen

LNSS04.F.7.1 Betextung möglich
Es ist möglich, eingefügte Abbildungen mit einem Text zu versehen.

LNSS04.F.7.2 Abbildungsverzeichnis
Es ist möglich, ein Abbildungsverzeichnis zu erstellen.

LNSS04.F.7.3 Numerierung
Es ist möglich, Abbildungen automatisch fortlaufend zu numerieren.

LNSS04.F.8 Dokumentenüberarbeitung

LNSS04.F.8.1 Kommentierung
Es ist möglich, ein erstelltes Dokument zu kommentieren und zu überarbeiten.

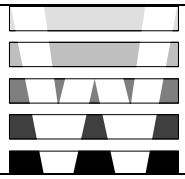
LNSS04.F.8.2 Kennzeichnung
Es ist möglich, Kommentare und Änderungen besonders zu kennzeichnen.

Die Kennzeichnung kann z. B. aus einer Linie am Seitenrand bestehen.

LNSS04.F.9 Druck

LNSS04.F.9.1 Komplettes Dokument
Der Ausdruck der Dokumentation ist mit einem Befehl möglich.

Der Befehl kann auch durch eine Stapeldatei oder durch ein Makro realisiert sein.



LNSS04.F.9.2 Teilausdruck
Es ist möglich anzuordnen, welcher Teil der Dokumentation gedruckt werden soll.
Die Seitenzahlen der Anfangs- und der Endseite können angegeben werden.

LNSS04.F.9.3 Selektiver Druck
Es ist möglich, die erzeugten Dokumente in unterschiedlichen Detaillierungsstufen oder für unterschiedliche Zielgruppen auszudrucken.

LNSS04.F.9.4 Verborgene Informationen
Es ist möglich, zum Zeitpunkt des Ausdruckes zu entscheiden, ob „verborgen“ formatierte Informationen gedruckt werden oder nicht.

LNSS04.F.10 Querverweise
Querverweise auf Seitenzahlen, Kapitel- und Abbildungsnummern werden automatisch verwaltet.
Ändert sich die Seitenzahl, an der ein definierter Text gedruckt wird, oder ändert sich die Nummer eines Kapitels oder einer Abbildung, so werden Bezüge hierauf automatisch entsprechend geändert.

LNSS04.F.11 Automatischer Nachvollzug von Änderungen
Änderungen von Grafiken und Texten werden automatisch in allen anderen Verwendungen nachvollzogen.

LNSS04.F.12 Bildschirmabzüge
Es ist möglich, Bildschirmabzüge einzufügen.

LNSS04.F.13 Variable Größe
Es ist möglich, die Größe einer eingefügten Abbildung zu verändern.
Der Grafikinhalt wird entsprechend der geänderten Größe angepaßt.

LNSS04.F.14 Zu integrierende Dokumente

LNSS04.F.14.1 Einfache Auswahl
Es ist möglich, zusammenzufügende Texte oder Grafiken einfach aus einer Liste oder ähnlichem auszuwählen.

LNSS04.F.14.2 Mehrfachverwendung
Es ist möglich, einen Text oder eine Grafik in mehrere Dokumente zu integrieren.

3.3.4.3.4 Sonstige Anforderungen

LNSS04.So.1 Makros
Es gibt eine Makrosprache, mit der Befehlsreihenfolgen zu einem Makro zusammengefaßt werden können.

LNSS04.So.2 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LNSS04.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.3.5 Leistungseinheit LNSS05 „Texte editieren“

3.3.5.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.3.5.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Bearbeitung von Texten gestellt.

3.3.5.3 Anforderungen

3.3.5.3.1 Anforderungen an die Schnittstellen

LNSS05.S.1 Ausgangs-Schnittstelle zu LNSS04 „Dokumentation aufbereiten“
Es ist möglich, editierte Texte ohne weitere Transformation über die Objektverwaltung an LNSS04 zur Aufbereitung der Dokumentation zu übergeben.

3.3.5.3.2 Anforderungen an die Methodenunterstützung

keine

3.3.5.3.3 Anforderungen an die Funktionen

LNSS05.F.1 Schreibmarke
Im Text ist durch eine Schreibmarke („Cursor“) erkenntlich, wo sich eine Operation (Texteingabe, Löschen, usw.) auswirken wird.

LNSS05.F.2 Bearbeitungsmodus

LNSS05.F.2.1 Einfügen
Es ist möglich, Text an der momentanen Position der Schreibmarke einzufügen.

LNSS05.F.2.2 Überschreiben
Es ist möglich, vorhandenen Text zu überschreiben.

LNSS05.F.3 Bewegen im Text

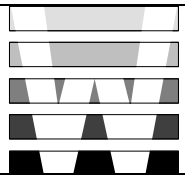
LNSS05.F.3.1 Buchstabenweise
Es ist möglich, mit der Schreibmarke genau einen Buchstaben vor oder zurück zu springen.

LNSS05.F.3.2 Wortweise
Es ist möglich, mit der Schreibmarke genau ein Wort vor oder zurück zu springen.

LNSS05.F.3.3 Zeilenweise
Es ist möglich, mit der Schreibmarke genau eine Zeile vor oder zurück zu springen.

LNSS05.F.3.4 Seitenweise
Es ist möglich, mit der Schreibmarke genau eine Seite vor oder zurück zu springen.
Mit „Seite“ ist dabei der am Bildschirm sichtbare Text gemeint.

LNSS05.F.3.5 Definierte Stellen
Es ist möglich, vordefinierte Stellen anzuspringen.



Vordefinierte Stellen können der Textanfang sowie das Textende sein. Bei mehrseitigen Texten sollte beispielsweise an den Beginn einer Druckseite oder den Beginn eines neuen Kapitels gesprungen werden können.

LNSS05.F.3.6 Blättern ohne Bewegen der Schreibmarke
Es ist möglich, im Text zu blättern, ohne die Schreibmarke zu bewegen.

Bei erneuter Nutzung der Tastatur wird mit der Texteingabe an der zuletzt aktiven Position fortgefahren.

LNSS05.F.4 Markieren von Text
Es ist möglich, eine Markierung an beliebiger Stelle im Text beginnen und enden zu lassen.

LNSS05.F.5 Löschen, kopieren, verschieben
Es ist möglich, markierten Text zu löschen oder an eine andere Stelle zu kopieren oder zu verschieben.

LNSS05.F.6 Mehrfachverwendung
Es ist möglich, auf bereits vorhandenen Text zuzugreifen, um gewünschte Teile zu kopieren.

„Vorhandener“ Text wurde zu einem früheren Zeitpunkt bearbeitet und abgespeichert.

LNSS05.F.7 Textausrichtung

LNSS05.F.7.1 Zeilenbreite
Es ist möglich, die Breite einer Zeile festzulegen.

LNSS05.F.7.2 Automatischer Zeilenumbruch
Bei Erreichen der vorgegebenen Zeilenbreite bricht der Text automatisch in die nächste Zeile um.

LNSS05.F.7.3 Feste Leerstellen
Es ist möglich, Leerstellen so zu kennzeichnen, daß an ihnen kein Zeilenumbruch erfolgt.

LNSS05.F.7.4 Absätze
Es ist möglich, zusammenhängenden Text als Absatz einzugeben.

LNSS05.F.7.5 Zeilenschaltung
Innerhalb eines Absatzes ist es möglich, eine gewollte Zeilenschaltung einzufügen.

Die neue Zeile beginnt am definierten Zeilenanfang, der Text gehört jedoch noch zum aktuellen Absatz.

LNSS05.F.7.6 Tabulatoren
Es ist möglich, beliebig viele links–, rechtsbündige, zentrierte oder Dezimaltabulatoren zu setzen.

Tabulatoren dienen der gleichmäßigen Ausrichtung von Texten mehrerer Zeilen.

LNSS05.F.8 Suchfunktion

LNSS05.F.8.1 Suchen
Es ist möglich, den Text nach vorgegebenen Zeichenfolgen zu durchsuchen.

LNSS05.F.8.2 Suchen und Ersetzen
Es ist möglich, den Text nach vorgegebenen Zeichenfolgen zu durchsuchen und die vorgegebenen Zeichenfolgen durch andere ersetzen zu lassen.

LNSS05.F.9 Automatischer Nachvollzug von Änderungen
Änderungen an Grafiken und Texten werden automatisch in allen anderen Verwendungen nachvollzogen.

3.3.5.3.4 Sonstige Anforderungen

LNSS05.So.1 Lesbarkeit
Der zu bearbeitende Text muß während der Texteingabe gut lesbar sein.
Gegebenenfalls muß in einen Anzeigemodus geschaltet werden, in dem der Text gut lesbar ist.

LNSS05.So.2 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LNSS05.So.3 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LNSS05.So.4 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.3.6 Leistungseinheit LNSS06 „Grafiken editieren“

3.3.6.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.3.6.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zum Editieren von Grafiken gestellt.

Die Verwendung von grafikorientierten Werkzeugen ist bei einer SEU in mehrfacher Hinsicht von Bedeutung:

- zur Methodenunterstützung bei bestimmten SEU-Werkzeugen (SE, PM) und
- zum freien Zeichnen, um beliebige Grafiken in Dokumente einzubringen (LNSS04).

Das grafische Editieren sollte daher sowohl das Einbinden und die methodenabhängige Manipulation vordefinierter Grafikelemente, als auch das pixel- und vektororientierte Zeichnen umfassen. Hier treffen ganz unterschiedliche Anforderungen aufeinander, nämlich die formfreie Erstellung von Grafiken – ähnlich dem Desktop Publishing – und die Manipulation von grafischen Objekten, die an bestimmte formale Methoden gebunden sind (z. B. Entity-Relationship-Symbolik). In diesem Sinne wird man zwischen zwei wesentlichen Varianten eines Grafikeditors unterscheiden: dem „freien“ Grafikeditor als Teil der Dokumentenbearbeitung (LNSS04), und der Gruppe der methodenunterstützenden Grafikeditoren, die direkt in die entsprechenden Werkzeuge integriert sein können.

3.3.6.3 Anforderungen

3.3.6.3.1 Anforderungen an die Schnittstellen

LNSS06.S.1 Eingangsschnittstelle zu LOBV03 „Datenstrukturen und Daten der SEU exportieren, importieren“
Es ist möglich, Grafiken, die mit externen Grafikeditoren erstellt und die mit LOBV03 importiert wurden, ohne weitere Transformation aus der Objektverwaltung zu übernehmen.

LNSS06.S.2 Ausgangs-Schnittstelle zu LNSS04 „Dokumentation aufbereiten“
Es ist möglich, editierte Grafiken ohne weitere Transformation über die Objektverwaltung an LNSS04 zur Aufbereitung der Dokumentation zu übergeben.

LNSS06.S.3 Konsistenz zwischen Grafiken und Datenbasis
Es ist sichergestellt, daß Grafiken und Objektverwaltung automatisch konsistent gehalten werden.
Änderungen der Diagramme werden in der Objektverwaltung nachgezogen und umgekehrt.

3.3.6.3.2 Anforderungen an die Methodenunterstützung

keine

3.3.6.3.3 Anforderungen an die Funktionen

LNSS06.F.1 Einfügen von Objekten

LNSS06.F.1.1 Einfügen neuer Objekte
Es ist möglich, neue Objekte einer Grafik hinzuzufügen.

LNSS06.F.1.2 Einfügen existierender Objekte
Es ist möglich, existierende Objekte einer Grafik hinzuzufügen.

LNSS06.F.2 Identifikation von Objekten

LNSS06.F.2.1 Benennung von Objekten
Es ist möglich, neue Objekte sowohl mit einem sprechenden Namen als auch mit einem Kurznamen zu versehen.

LNSS06.F.2.2 Umbenennung von Objekten
Es ist möglich, Objekte umzubenennen.

LNSS06.F.2.3 Eindeutigkeit von Namen
Beim Neuanlegen von Objekten werden bereits verwendete Namen für Objekte gleichen Typs zurückgewiesen.

LNSS06.F.3 Beschreibung von Objekten

LNSS06.F.3.1 Freitextfeld
Es ist möglich, Objekte mittels eines Freitextfeldes zu beschreiben.

LNSS06.F.3.2 Zugriff
Es ist möglich, auf Objektbeschreibungen über das jeweilige Symbol aus einer Grafik zuzugreifen.

LNSS06.F.3.3 Verknüpfung
Die Beschreibung ist mit dem Objekt verknüpft, so daß ein Löschen des Objektes das Löschen der Beschreibung nach sich zieht.

LNSS06.F.3.4 Kopieren in eine andere Objektbeschreibung
Es ist möglich, Text aus der Beschreibung eines Objektes in eine andere Objektbeschreibung zu kopieren.

LNSS06.F.4 Markieren von Objekten

LNSS06.F.4.1 Markieren einzelner Objekte
Es ist möglich, Objekte einzeln zu markieren.

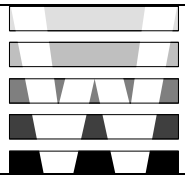
LNSS06.F.4.2 Markieren von Objektgruppen
Es ist möglich, Gruppen von Objekten zu markieren.

LNSS06.F.4.3 Explizite und implizite Markierung
Die Markierung ist möglich, indem die betroffenen Objekte explizit oder parametergesteuert aktiviert werden.

Parametergesteuerte Markierung erfolgt z. B. über den Objektnamen, über den grafischen Zusammenhang (unmittelbare Nachbarschaft eines bisher schon markierten Objekts), über Abhängigkeiten, usw.

LNSS06.F.4.4 Aufheben der Markierung
Es ist möglich, die Markierung von Objekten aufzuheben, indem die betroffenen Objekte explizit oder parametergesteuert deaktiviert werden.

Parametergesteuertes Aufheben von Markierungen erfolgt z. B. über den Objektnamen, über den grafischen Zusammenhang (unmittelbare Nachbarschaft eines bisher schon markierten Objekts), über Abhängigkeiten, usw.



LNSS06.F.5 Kopieren von markierten Objekten
Es ist möglich, markierte Objekte eines Diagramms in ein anderes Diagramm zu kopieren.

LNSS06.F.6 Löschen und Ausblenden von markierten Objekten

LNSS06.F.6.1 Löschen
Es ist möglich, markierte Objekte aus der Datenbasis zu löschen.

LNSS06.F.6.2 Konsistenzsicherung
Tritt ein zu löschendes Objekt in mehreren Kontexten auf, so wird eine entsprechende Meldung ausgegeben, und ein Löschen erfolgt nur nach vorhergehender Bestätigung.

LNSS06.F.6.3 Ausblenden
Es ist möglich, markierte Objekte vorübergehend aus einer Grafik auszublenden.

Ausgeblendete Objekte werden in der Datenbasis nicht gelöscht.

LNSS06.F.6.4 Zwischenablage
Es ist möglich, markierte Objekte aus der Grafik in eine Zwischenablage zu löschen oder zu kopieren und an einer anderen Stelle wiedereinzufügen.

LNSS06.F.6.5 Wahrung der Konsistenz grafischer Darstellungen
Beim Löschen bzw. Ausblenden von Objekten werden abhängige Objekte automatisch gelöscht, sofern sie nicht in anderen Kontexten auftreten.

Abhängige Objekte eines Prozesses sind beispielsweise ein- bzw. ausgehende Datenflüsse.

LNSS06.F.7 Plazieren von markierten Objekten

LNSS06.F.7.1 Verschieben
Es ist möglich, markierte Objekte zu verschieben, wobei die Verbindungen zu anderen Objekten automatisch so nachgeführt werden, daß die Proportionen gleich bleiben.

Verbindungskanten werden z. B. automatisch verkürzt oder verlängert.

LNSS06.F.7.2 Freie Plazierung
Es ist möglich, markierte Objekte frei in einer Grafik zu plazieren.

LNSS06.F.7.3 Plazierung mittels Raster
Es ist möglich, für die Plazierung markierter Objekte ein skalierbares Raster zu benutzen.

LNSS06.F.7.4 Speichern der gewählten Plazierung
Die gewählte Plazierung der Objekte in einer Grafik wird gespeichert, so daß beim nachfolgenden Anzeigen der Grafik die zuletzt festgelegte Plazierung der Objekte erhalten bleibt.

LNSS06.F.7.5 Vorder-/Hintergrund
Es ist möglich, markierte Objekte in den Bildschichten der Zeichenfläche in den Vorder- oder Hintergrund zu verlagern.

LNSS06.F.7.6 Ausrichtung
Es ist möglich, markierte Objekte in Grafiken nach bestimmten Kriterien (unten, Mitte, oben, zentriert, linker Rand, rechter Rand) auszurichten.

LNSS06.F.8 Spiegelung und Drehung von markierten Objekten
Es ist möglich, markierte Objekte zu spiegeln und zu drehen.

LNSS06.F.9 Skalierung von markierten Objekten
Es ist möglich, markierte Objekte zu skalieren.

LNSS06.F.10 Duplizieren von markierten Objekten
Es ist möglich, markierte Objekte zu duplizieren.

LNSS06.F.11 Komplexe Objekte

LNSS06.F.11.1 Erstellen komplexer Objekte
Es ist möglich, markierte Objekte zu einer Einheit zusammenzufassen, wobei diese Zusammenfassung von Objekten zu Einheiten mehrstufig erfolgen kann.

LNSS06.F.11.2 Auflösen komplexer Objekte
Es ist möglich, markierte Objekte, die vorher zu einer Einheit zusammengefaßt wurden, wieder in ihre ursprünglichen Bestandteile (gegebenenfalls mehrstufig) zu zerlegen.

LNSS06.F.11.3 Manipulation komplexer Objekte
Die Manipulationsmöglichkeiten für komplexe Objekte entsprechen denen einfacher Objekte.

LNSS06.F.12 Zeichenfläche

LNSS06.F.12.1 Größe und Format
Es ist möglich, Größe und Format der Zeichenfläche festzulegen.

LNSS06.F.12.2 Dynamische Erweiterung der Zeichenfläche
Die Größe der Zeichenfläche wird bei entsprechendem Platzbedarf automatisch dynamisch erweitert.

LNSS06.F.12.3 Rasterung der Zeichenfläche
Es ist möglich, die Zeichenfläche mit DIN A4- und DIN A3-Formaten zu rastern.

LNSS06.F.12.4 Hochformat und Querformat
Es ist möglich, die Rasterung der Zeichenfläche mit DIN A4- und DIN A3-Format sowohl im Hochformat als auch im Querformat durchzuführen.

LNSS06.F.12.5 Bildschichten
Es ist möglich, für die Zeichenfläche mehrere Bildschichten (Layer) zu definieren.

LNSS06.F.13 Vektorgrafik

LNSS06.F.13.1 Vordefinierte Symbole

LNSS06.F.13.1.1 Linien
Es ist möglich, Linien frei skalierbar zu zeichnen.

LNSS06.F.13.1.2 Rechtecke
Es ist möglich, Rechtecke frei skalierbar zu zeichnen.

LNSS06.F.13.1.3 Quadrate
Es ist möglich, Quadrate frei skalierbar zu zeichnen.

LNSS06.F.13.1.4 Kreise
Es ist möglich, Kreise frei skalierbar zu zeichnen.

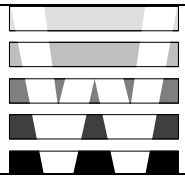
LNSS06.F.13.1.5 Ellipsen
Es ist möglich, Ellipsen frei skalierbar zu zeichnen.

LNSS06.F.13.1.6 Polygone
Es ist möglich, Polygone frei skalierbar zu zeichnen.

LNSS06.F.13.1.7 Kreissegmente
Es ist möglich, Kreissegmente frei skalierbar zu zeichnen.

LNSS06.F.13.1.8 Liniendicke und Linienart
Es ist möglich, für alle Symbole die Liniendicke und die Linienart zu ändern.

LNSS06.F.13.2 Nutzerdefinierte Symbole
Es ist möglich, neue, nutzerdefinierte Symbole in den Symbolvorrat aufzunehmen.



LNSS06.F.14 Geschäftsgrafiken
Es ist möglich, numerische Datenbeziehungen mit Hilfe von Balken- oder Kreissegment-Diagrammen darzustellen.

LNSS06.F.15 Freihandgrafik

LNSS06.F.15.1 Zeichenmittel
Es sind „Malstift“, „Pinsel“ und „Spraydose“ vorhanden.

LNSS06.F.15.2 Stärken und Formen
Für alle Zeichenmittel sind verschiedene Stärken und Formen verfügbar.

LNSS06.F.15.3 Radiergummi
Es ist ein „Radiergummi“ vorhanden.

LNSS06.F.16 Grafikmodus

LNSS06.F.16.1 Pixelgrafik und Vektorgrafik
Es ist möglich, sowohl im Pixelgrafik- als auch im Vektorgrafikmodus zu zeichnen.

LNSS06.F.16.2 Rasterdarstellung für Pixel
Für das Zeichnen im Pixelmodus ist eine vergrößerte Rasterdarstellung verfügbar.

LNSS06.F.16.3 Verbindung von Grenzpunkten
Es ist möglich, die Schreibmarke beim Zeichnen von Vektoren auf Wunsch automatisch so zu positionieren, daß Grenzpunkte verbunden sind, d. h. zwei angrenzende Striche sind somit automatisch verknüpft und bilden eine Ecke („Align“).

LNSS06.F.17 Optik und Layout

LNSS06.F.17.1 Text
Es ist möglich, in eine Grafik frei definierbaren Text einzufügen.

LNSS06.F.17.2 Zeichensätze
Es ist möglich, Texte mit verschiedenen Schrift-Zeichensätzen in skalierbaren Größen darzustellen.

LNSS06.F.17.3 Textmodifikation
Es ist möglich, Text in einem komplexen Objekt zu modifizieren, ohne vorher das komplexe Objekt in seine Bestandteile zerlegen zu müssen.

LNSS06.F.17.4 Füllmuster
Es ist möglich, begrenzte Flächen mit wählbaren Mustern oder Farben automatisch zu füllen.

LNSS06.F.17.5 Automatische Texttrennung
Es ist möglich, Beschriftungen automatisch umgebungskonform zu trennen.

LNSS06.F.17.6 Automatische Layout-Verbesserungen
Es ist möglich, automatisch Layout-Verbesserungen von Grafiken durchzuführen.

LNSS06.F.18 Manipulation gesamter Grafiken

LNSS06.F.18.1 Löschen
Es ist möglich, Grafiken zu löschen.

LNSS06.F.18.2 Skalieren
Es ist möglich, Grafiken beliebig zu vergrößern und verkleinern.

LNSS06.F.18.3 Invertieren
Es ist möglich, Grafiken zu invertieren.

LNSS06.F.18.4 Rotieren
Es ist möglich, Grafiken zu rotieren.

LNSS06.F.19 Ausgabe

LNSS06.F.19.1 Bildschirm
Es ist möglich, Grafiken am Bildschirm anzuzeigen.

LNSS06.F.19.2 Drucker
Es ist möglich, Grafiken auszudrucken.

LNSS06.F.19.3 Druckformate und Auflösungen
Es ist möglich, die Grafiken für verschiedene Druckformate oder Auflösungen automatisch einzurichten.

Z. B. werden Überkreuzungen von Linien minimal gehalten.

LNSS06.F.19.4 Druck über mehrere Seiten
Es ist möglich, Grafiken über mehrere Seiten so auszudrucken, daß die einzelnen Seiten schnittgenau zusammenpassen.

LNSS06.F.19.5 Umleiten Druckauftrag
Es ist möglich, einen Druckauftrag statt auf den Drucker auf eine Datei umzuleiten.

LNSS06.F.20 Automatischer Nachvollzug von Änderungen
Änderungen an Grafiken und Texten werden automatisch in allen anderen Verwendungen nachvollzogen.

3.3.6.3.4 Sonstige Anforderungen

LNSS06.So.1 Komprimierung
Es ist möglich, Grafiken komprimiert zu speichern.

Dies ist insbesondere für Pixelgrafiken sinnvoll.

LNSS06.So.2 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LNSS06.So.3 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LNSS06.So.4 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.4 Leistungskomplex „Systementwicklung steuern“

Dieser Leistungskomplex ist nicht weiter unterteilt und wird somit als gleichnamige Leistungseinheit übernommen.

3.4.1 Leistungseinheit LSES01 „Systementwicklung steuern“

3.4.1.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.4.1.2 Kurzcharakteristik

Mit dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen das V-Modell auf den Abstraktionsebenen

- bereichsspezifisches und
- projektspezifisches V-Modell

IT-unterstützt nachgebildet werden kann.

Es kann durch das Aktivitäten-Produkte-Netz navigiert werden. Dabei ist automatisch gesichert, daß eine vom Ablauf der Systementwicklung her sinnvolle, also in sich konsistente Bearbeitungsreihenfolge der Aktivitäten strikt eingehalten wird.

Die übrigen Werkzeuge werden kontextabhängig verfügbar gemacht. Es werden diejenigen Produkte bereitgestellt, die mit diesen Werkzeugen weiterbearbeitet werden sollen.

3.4.1.3 Anforderungen

3.4.1.3.1 Anforderungen an die Schnittstellen

LSES01.S.1 Steuerungsschnittstelle
Es ist möglich, über offengelegte, dokumentierte Schnittstellen Steuerungsparameter mit anderen Werkzeugen auszutauschen.

LSES01.S.2 Eingangs-Schnittstelle zu LPM03 „Ausschreibungsrelevantes Tailoring unterstützen“
Es ist möglich, mit LPM03 erfaßte projektspezifische V-Modelle ohne weitere Transformation aus der Objektverwaltung für die Steuerung der Systementwicklung zu übernehmen.

LSES01.S.3 Eingangs-Schnittstelle zu LPM04 „Planung des Projektablaufs unterstützen“
Es ist möglich, mit LPM04 erfaßte Ausprägungen von Aktivitätentypen und Ressourcenzuordnungen aus Projektablaufplänen ohne weitere Transformation aus der Objektverwaltung für die Steuerung der Systementwicklung zu übernehmen.

3.4.1.3.2 Anforderungen an die Methodenunterstützung

keine

3.4.1.3.3 Anforderungen an die Funktionen

LSES01.F.1 Darstellungsmöglichkeiten

LSES01.F.1.1 Überblick
Es ist möglich, einen Gesamtüberblick über die Aktivitäten und Produkte des V-Modells darzustellen.

LSES01.F.1.2 Position
Zu jedem Zeitpunkt ist im Projektablauf darstellbar, wo man sich bezogen auf den Gesamtablauf befindet.

LSES01.F.1.3 Zustände
Es ist darstellbar, welche Produkte sich in welchem Zustand befinden.

LSES01.F.1.4 Bearbeitungsstand
Es ist möglich, den Bearbeitungsstand aller Produkte kapitelweise darzustellen, soweit die Erarbeitung der Kapitel in unterschiedlichen Aktivitäten erfolgt.

LSES01.F.2 Projektkontrolle

LSES01.F.2.1 Plankonsistenz
Es ist möglich, die Konsistenz mit dem Projektplan und den Prüfplänen zu prüfen.

LSES01.F.2.2 Rollenänderung
Es ist möglich, die Rollenverteilung im laufenden Projekt zu ändern.

LSES01.F.2.3 Positionsverwaltung
Die nutzerspezifische Position im Gesamtnetz aller Aktivitäten und Produkte wird verwaltet.

LSES01.F.3 V-Modell-Unterstützung

LSES01.F.3.1 Strukturabbildung
Es ist möglich, die Strukturen des V-Modells und der Methodenzuordnung vollständig und ohne Einschränkungen abzubilden.

LSES01.F.3.2 Nutzer und Rolle
Es ist möglich, zwischen Rollen und Nutzern zu unterscheiden. Die Rollenzuordnung im Sinne des V-Modells ist möglich.

LSES01.F.3.3 Aktivitätenfolge
Die Umgehung der Ausführungsreihenfolge einzelner Aktivitäten gemäß V-Modell ist nur bei expliziter und automatisch dokumentierter Intervention möglich.

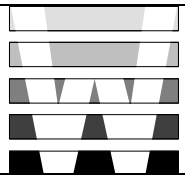
LSES01.F.3.4 Modelländerung
Es ist möglich, das projektspezifische V-Modell im laufenden Projekt zu ändern.

LSES01.F.3.5 Navigation
Es ist möglich, abhängig vom Zustand der Produkte im Aktivitäten- und Produkt-Netz zu navigieren.

LSES01.F.4 Zusammenarbeit mit anderen Werkzeugen

LSES01.F.4.1 Direkteinbindung
Die direkte Einbindung von Werkzeugen ist möglich.

LSES01.F.4.2 Werkzeugaufruf
Der Aufruf von Werkzeugen ist ausschließlich über LSES01 möglich.



LSES01.F.5 Aktivitätenbehandlung

LSES01.F.5.1 Historie
Die Aktivitätenhistorie ist nutzerspezifisch darstellbar.
Es ist darstellbar, welche Aktivität wann durch wen bearbeitet wurde.

LSES01.F.5.2 Alternativen
Es ist möglich, Alternativen der Aktivitätenfolge darzustellen.

LSES01.F.6 Teamfähigkeit
Es ist möglich, daß gleichzeitig mehrere Nutzer innerhalb eines Projektes unterstützt werden.

3.4.1.3.4 Sonstige Anforderungen

keine

3.5 Leistungskomplex „Sicherheit“

Hinweis:

Nutzer und Prozesse werden im folgenden unter dem Begriff „Subjekt“ zusammengefaßt.

3.5.1 Leistungseinheit LSIC01 „Allgemeine Sicherheitsprinzipien“

3.5.1.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.5.1.2 Kurzcharakteristik

Diese Leistungseinheit beschreibt die allgemeinen Prinzipien, die durch den Leistungskomplex Sicherheit abgedeckt werden. Diese prinzipiellen Anforderungen leiten sich großenteils (sekundäre Sicherheitsanforderungen) aus den (primären) Sicherheitsanforderungen der übrigen Leistungseinheiten im Leistungskomplex Sicherheit ab. Sekundäre Sicherheitsanforderungen gewährleisten den Schutz der Sicherheitsmechanismen und sind deshalb zur Erlangung der angestrebten Sicherheit unverzichtbar.

3.5.1.3 Anforderungen

3.5.1.3.1 Anforderungen an die Schnittstellen

keine

3.5.1.3.2 Anforderungen an die Methodenunterstützung

keine

3.5.1.3.3 Anforderungen an die Funktionen

LSIC01.F.1 Abgeleitete Sicherheitsanforderungen

LSIC01.F.1.1 Unumgehbarkeit
Die Sicherheitsmechanismen sind so in das Gesamtsystem integriert, daß sie nicht umgangen werden können.

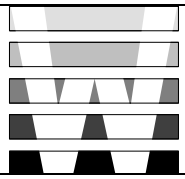
LSIC01.F.1.2 Nichtausschaltbarkeit
Für unprivilegierte Nutzer besteht keine Möglichkeit, die Sicherheitsmechanismen auszuschalten.

LSIC01.F.1.3 Untäuschbarkeit
Die Sicherheitsmechanismen können nicht getäuscht werden.

LSIC01.F.1.4 Manipulationssicherheit
Es sind keine Manipulationen der Sicherheitsmechanismen möglich.

LSIC01.F.1.5 Korrektheit
Die Sicherheitsmechanismen sind so strukturiert und überschaubar, daß sie vollständigen Analysen und umfassenden Tests unterzogen werden können.

LSIC01.F.1.6 Separierung
Die nicht vertrauenswürdigen Komponenten der SEU sind von den vertrauenswürdigen so separiert, daß durch sie keine sicherheitsrelevanten Aktionen durchgeführt oder beeinflußt werden können.



LSIC01F.2 Systemweite eindeutige Identifikationsnummer jedes Objektes
Jedes Objekt hat eine eindeutige Identifikationsnummer. Die Identifikationsnummern sind fortlaufend.

Mittels der Identifikationsnummer kann u. a. der Verlust oder das unbefugte Wiedereinspielen eines Objektes entdeckt werden.

LSIC01.F.3 Protokollierbarkeit
Die Komponenten der SEU sind in der Lage, die für eine Aufzeichnung sicherheitsrelevanter Ereignisse erforderlichen Daten an die Leistungseinheit LSIC04 „Auditing“ zu übergeben, oder der für die Aufzeichnung erforderliche Code muß in den entsprechenden vertrauenswürdigen Komponenten integriert sein.

3.5.1.3.4 Sonstige Anforderungen

keine

3.5.2 Leistungseinheit LSIC02 „Identifikation und Authentisierung“

3.5.2.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.5.2.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen zusammengefaßt, die die Bestimmung und Überwachung derjenigen Nutzer unterstützen, die Zugriff auf Betriebsmittel haben, welche von der SEU kontrolliert werden, insbesondere die Produktbibliothek.

Die Leistungseinheit umfaßt alle Funktionen, die dazu beitragen

- die behauptete Identität eines Subjekts zu bestimmen und nachzuprüfen,
- unberechtigte Zugangsversuche abzuweisen,
- neue Nutzeridentitäten hinzuzufügen, alte herauszunehmen oder ungültig zu machen,
- Authentisierungsinformationen zu erzeugen, zu ändern oder abzufragen,
- die Integrität der Authentisierungsinformationen zu gewährleisten,
- die unbefugte Verwendung der Authentisierungsinformationen zu verhindern,
- wiederholte Versuche zur Erlangung einer falschen Identität zu unterbinden.

3.5.2.3 Anforderungen

3.5.2.3.1 Anforderungen an die Schnittstellen

LSIC02.S.1 Schnittstelle zu LNSS01 „Funktionen der Nutzeroberfläche“
Die Identifikation und Authentisierung kann über die Leistungseinheit „Funktionen der Nutzeroberfläche“ sowohl durch den Nutzer selbst (z. B. Login-Prozedur) als auch durch die SEU (z. B. nach längerer Inaktivität des Nutzers) angestoßen werden.

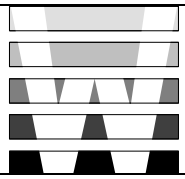
LSIC02.S.2 Schnittstelle zu LSIC03 „Zugriffskontrolle“

LSIC02.S.2.1 Schutz der Identifikations- und Authentisierungsdaten
Die Leistungseinheit „Zugriffskontrolle“ bietet die Funktionalität, daß der Mechanismus Identifikation/Authentisierung als „geschütztes Subsystem“ realisiert werden kann. Das bedeutet, daß die Nutzer nur über vordefinierte Funktionen (z. B. unter UNIX das Kommando passwd zur Aktualisierung des eigenen Paßwortes) auf die Identifikations- und Authentisierungsdaten zugreifen können.

LSIC02.S.2.2 Verfügbarmachen der Identifikations- und Authentisierungsdaten
Die Leistungseinheit LSIC03 setzt einen korrekt funktionierenden, unumgehbaren Mechanismus zur Identifikation und Authentisierung voraus.Die Leistungseinheit „Identifikation und Authentisierung“ stellt der Leistungseinheit LSIC03 alle zur Verwaltung und Kontrolle von Zugriffsrechten erforderlichen Identifikations- und Authentisierungsdaten zur Verfügung.

Dazu zählen u. a. Informationen über die Ermächtigung und Berechtigung einzelner Nutzer.

LSIC02.S.3 Schnittstelle zu LSIC04 „Auditing“
Der Mechanismus zur Identifizierung und Authentisierung von Nutzern erlaubt es, berechtigte Nutzer eindeutig gegenüber der Leistungseinheit „Auditing“ zu identifizieren.



Bei jeder Authentisierung, ob erfolgreich oder erfolglos, werden die für die Protokollierung dieses Ereignisses notwendigen Informationen (u. a. Parameter, Erfolg bzw. Mißerfolg des Ereignisses) an die Leistungseinheit „Auditing“ übergeben.

LSIC02.S.4 Schnittstelle zu LSIC06 „Unverfälschtheit, Vertraulichkeit und Verbindlichkeit“
Die Authentisierungsdaten werden nur verschlüsselt gespeichert oder übertragen. Die Leistungseinheit „Unverfälschtheit, Vertraulichkeit und Verbindlichkeit“ setzt dabei ein sicheres Einweg-Verfahren ein.

Dadurch werden gezielte Fälschungen von Authentisierungsdaten verhindert. Es ist gewährleistet, daß generell kein Einblick in die Authentisierungsdaten (z. B. Paßwörter) erfolgen kann – auch nicht durch den Systemverwalter. U. a. wird auch verhindert, daß ein Eindringling durch Abhören einer Verbindungsleitung (z. B. zwischen Terminal und Rechner) oder durch lesenden Zugriff auf die Paßwortdatei in den Besitz eines unverschlüsselten Paßwortes gelangt.

3.5.2.3.2 Anforderungen an die Methodenunterstützung

keine

3.5.2.3.3 Anforderungen an die Funktionen

LSIC02.F.1 Mechanismen

LSIC02.F.1.1 Existenz
Es existiert ein Mechanismus zur Identifikation und Authentisierung von sowohl lokalen als auch entfernten Nutzern.

LSIC02.F.1.2 Eindeutigkeit
Es ist möglich, Nutzer eindeutig zu identifizieren und authentisieren.

LSIC02.F.1.3 Authentifikation durch Wissen

LSIC02.F.1.3.1 Paßwortlänge
Es wird erzwungen, daß Paßwörter eine Mindestlänge besitzen. Diese ist durch den Systemverwalter konfigurierbar (untere Grenze sind 6 Zeichen).

LSIC02.F.1.3.2 Komplexität
Es wird erzwungen, daß jedes Paßwort mindestens ein Sonderzeichen und eine Ziffer enthält.

LSIC02.F.1.3.3 Nicht-triviale Paßwörter
Triviale Paßwörter, d. h. solche, die aus der Nutzerinformation des zugehörigen Nutzers ableitbar sind (z. B. Nutzername, Vorname, Rechnername), werden nicht akzeptiert.

LSIC02.F.1.3.4 Schutz vor verdeckten Kanälen
Auch bei Eingabe einer ungültigen Identifikation wird nicht sofort abgebrochen, sondern erst nach der Eingabe eines Paßworts.

Dadurch kann ein Eindringling nicht erkennen, ob die Identität gültig und nur das Paßwort falsch war, oder ob bereits die Identifikation ungültig war.

LSIC02.F.1.3.5 Verdeckte Eingabe
Das eingegebene Paßwort wird nicht angezeigt.

LSIC02.F.1.3.6 Paßworterneuerung
Regelmäßige Paßwortänderungen werden vom System erzwungen. Die Zeiträume dafür sind festlegbar.

LSIC02.F.1.3.7 Historie
Das System speichert über eine gewisse Zeit alle gewählten Paßwörter pro Nutzerkennung und verhindert, daß innerhalb dieses Zeitraumes ein Paßwort mehrmals ausgewählt wird.

Damit wird verhindert, daß ein Nutzer zwar formal sein Paßwort erneuert, daß aber das neue identisch mit dem alten ist und somit der eigentliche Zweck der Paßworterneuerung unterlaufen wird.

LSIC02.F.1.3.8 Ausschlußliste
Es ist möglich, in einer Ausschlußliste diejenigen Paßwörter anzugeben, die vom System nicht akzeptiert werden sollen.

LSIC02.F.1.4 Authentifikation durch Besitz

LSIC02.F.1.4.1 Präsenz
Es wird das Vorhandensein eines maschinenlesbaren Ausweises geprüft.

LSIC02.F.1.4.2 ID-Kennzeichen
Es wird überprüft, ob der Ausweis eine eindeutige und gültige ID-Kennung enthält.

LSIC02.F.1.4.3 Kennzahl
Das System vergibt für jeden Ausweis eine individuelle Kennzahl (PIN). Diese Kennzahl ist mindestens sechsstellig.

LSIC02.F.1.4.4 Systemzugang
Der Systemzugang wird nur gewährt, wenn ein maschinenlesbarer Ausweis vorgelegt wurde, der eine eindeutige und gültige ID-Kennung enthält, und zusätzlich vom Nutzer die richtige Kennzahl (PIN) eingegeben wurde.

LSIC02.F.1.4.5 Verdeckte Eingabe
Die vom Nutzer eingegebene Kennzahl wird nicht angezeigt.

LSIC02.F.1.5 Authentifikation durch persönliche Merkmale
Charakterisierende Eigenschaften (z. B. Fingerabdruck, Augenhintergrund, Schreibgeschwindigkeit) werden als Authentisierungsdaten gespeichert und bei einer Authentisierung auf Übereinstimmung mit der aktuellen Eingabe geprüft.

LSIC02.F.2 Verwaltung der Identifikations- und Authentisierungsdaten

LSIC02.F.2.1 Nutzerverwaltung
Es besteht für den Systemverwalter die Möglichkeit, neue Nutzer anzulegen sowie alte zu löschen oder zu sperren.

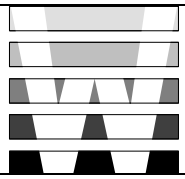
LSIC02.F.2.2 Identifikationsdaten
Es besteht für den Systemverwalter die Möglichkeit, im Rahmen der Systemverwaltung die Identifikationsdaten einzusehen.

Dazu zählen beispielsweise die Nutzerkennung und die Nutzer-ID.

LSIC02.F.3 Trennung von Identifikations- und Authentisierungsdaten
Die Authentisierungsdaten sind von den Identifikationsdaten separiert und in einer eigenen Datei gespeichert, auf die nur der Systemverwalter Zugriff hat.

Der Systemverwalter muß schreibenden Zugriff auf die Authentisierungsdaten besitzen, u. a. um neue Nutzer (mit einem initialen Paßwort) einrichten zu können, oder einem Nutzer, der sein Paßwort vergessen hat, den berechtigten Zugang zu gewähren und die Möglichkeit zu geben, ein neues Paßwort zu vereinbaren.

LSIC02.F.4 Zugangsschutz
Es wird erzwungen, daß erst nach erfolgreicher Identifikation und Authentisierung eines Nutzers dieser weitere Interaktionen durchführen kann.



LSIC02.F.5 Wiederholungssperre
Bei mehrmaligem Fehlversuch zur Identifikation und Authentisierung eines bestimmten Nutzers wird dessen Nutzerkennung gesperrt. Es kann festgelegt werden, wieviele Wiederholungen gestattet sein sollen.

LSIC02.F.6 Vertrauenwürdiger Pfad
Die Identifikation und Authentisierung erfolgt grundsätzlich über einen vertrauenswürdigen Pfad zwischen Nutzer und System. Kommunikationen über diesen Pfad können ausschließlich durch den betreffenden Nutzer selbst oder die vertrauenswürdigen Komponenten der SEU initiiert werden und sind von anderen Pfaden logisch isoliert und unmißverständlich unterscheidbar.

Dadurch ist sichergestellt, daß die Authentisierung tatsächlich durch die SEU und nicht durch ein (nicht vertrauenswürdiges) Anwendungsprogramm erfolgt. Der vertrauenswürdige Pfad wird immer dann benutzt, wenn eine vertrauenswürdige SEU-Nutzer-Verbindung erforderlich ist. Das kann z. B. beim Login oder bei einer Änderung der aktuellen Sicherheitsstufe eines Subjekts sein.

LSIC02.F.7 Identifizierbarkeit
Bei jeder Interaktion ist es möglich, die Identität des Nutzers festzustellen.

LSIC02.F.8 Partner-Authentisierung
Vor jeder Herstellung einer Kommunikationsverbindung wird der Kommunikationspartner (Rechner, Prozeß oder Nutzer) eindeutig identifiziert und authentisiert. Erst nach erfolgreicher Identifikation und Authentisierung kann eine Übertragung der Nutzdaten erfolgen.

LSIC02.F.9 Sender-Authentisierung
Beim Empfang von Nachrichten, wird deren Absender eindeutig identifiziert und authentisiert.

LSIC02.F.10 Endgerätezuordnung
Es ist möglich, einzelnen Nutzern bestimmte Endgeräte zuzuordnen, von welchen aus sie sich einloggen dürfen. Login-Versuche von anderen Endgeräten aus werden abgewiesen.

LSIC02.F.11 Individueller Zeitraum
Es ist möglich, jedem Nutzer individuelle Zeiträume zuzuordnen, innerhalb derer er Login-Prozesse starten darf. Außerhalb dieser Zeiträume werden seine Login-Versuche grundsätzlich abgewiesen.

LSIC02.F.12 Login-Anzeige
Bei erfolgreichem Login-Vorgang wird dem Nutzer Datum und Uhrzeit seines letzten Login-Vorgangs angezeigt.

LSIC02.F.13 Erneute Authentisierung

LSIC02.F.13.1 Manuell aktivierbarer Zugangsschutz
Jeder Nutzer kann eine Bildschirm-Dunkelschaltung des eigenen Rechners auslösen (z. B. durch eine „Pausetaste“). Die Arbeit kann erst nach einer erneuten Identifikation und Authentisierung fortgesetzt werden.

LSIC02.F.13.2 Automatisch aktivierter Zugangsschutz
Es besteht die Möglichkeit, erneute Authentisierungen zu erzwingen.

Erfolgen z. B. längere Zeit keine Eingaben des Nutzers, so kann er seine Aktivitäten erst nach einer erneuten Identifikation und Authentisierung wieder aufnehmen. Die Zeit, nach welcher der Zugangsschutz aktiviert wird, ist konfigurierbar. Außerdem können neben dem Login weitere Aktionen (z. B. hochsensitive) festgelegt werden, vor deren Ausführung eine zusätzliche Identifikation und Authentisierung erforderlich ist.

3.5.2.3.4 Sonstige Anforderungen

keine

3.5.3 Leistungseinheit LSIC03 „Zugriffskontrolle“

3.5.3.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität

- KM 1.2 KM einrichten
- KM 2.5 Zugriffsrechte verwalten

Methode

keine

3.5.3.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen zusammengefaßt, die sicherstellen, daß keine Subjekte, d. h. Nutzer oder Prozesse im Auftrag von Nutzern, Zugriff auf Informationen oder Betriebsmittel erhalten, für die sie kein Zugriffsrecht haben oder für die keine Notwendigkeit für einen Zugriff besteht (need-to-know). Ebenfalls enthalten sind Anforderungen, die das unbefugte Erzeugen oder Ändern (einschließlich Löschen) von Informationen unterbinden.

Darin eingeschlossen sind alle Funktionen, die für die Kontrolle

- des Informationsflusses zwischen Nutzern, Prozessen und Objekten,
- der Betriebsmittelnutzung durch Nutzer, Prozesse und Objekte

vorgesehen sind. Das schließt die Verwaltung (z. B. Vergabe und Zurücknahme) von Zugriffsrechten und ihre Prüfung ein.

Die Leistungseinheit umfaßt alle Funktionen, die

- zur Erstellung und Fortschreibung von Listen oder Regeln, durch die die gültigen Zugriffsrechte definiert sind,
- zur zeitweiligen Einschränkung von Zugriffen auf Objekte, auf die mehrere Nutzer oder Prozesse gleichzeitig Zugriff haben,
- zur Erhaltung der Konsistenz und Integrität solcher Objekte,
- zur Sicherstellung, daß Objekten bei ihrer Erzeugung voreingestellte Zugriffslisten oder -regeln zugeordnet werden,
- zur Kontrolle der Weitergabe von Zugriffsrechten,
- zur Überwachung einer Ableitung von geschützten Informationen, die durch Zusammenfügen von legitim zugänglichen Daten möglich ist,

vorgesehen sind.

3.5.3.3 Anforderungen

3.5.3.3.1 Anforderungen an die Schnittstellen

LSIC03.S.1 Schnittstelle zu LOBV01 „Datenstrukturen und Daten der SEU erstellen, speichern, verwalten“
Alle Zugriffsversuche auf Objekte, die der Objektverwaltung unterliegen, werden zunächst der Zugriffskontrolle unterzogen.

LSIC03.S.2 Schnittstelle zu LOBV02 „Datenstrukturen und Daten der SEU auswerten, abfragen“
Alle Zugriffsversuche auf Objekte, die der Objektverwaltung unterliegen, werden zunächst der Zugriffskontrolle unterzogen.

LSIC03.S.3 Schnittstelle zu LOBV03 „Datenstrukturen und Daten der SEU exportieren, importieren“
Beim Import und Export von Objekten werden die Zugriffsberechtigungen berücksichtigt.

LSIC03.S.4 Schnittstelle zu LSIC02 „Identifikation und Authentisierung“

LSIC03.S.4.1 Schutz der Identifikations- und Authentisierungsdaten
Die Leistungseinheit „Zugriffskontrolle“ stellt sicher, daß nur der Systemverwalter auf die Identifikations- und Authentisierungsdaten zugreifen kann.

LSIC03.S.4.2 Verfügbarmachen der Identifikations- und Authentisierungsdaten
Die Leistungseinheit „Zugriffskontrolle“ setzt einen korrekt funktionierenden, unumgehbaren Mechanismus zur Identifikation und Authentisierung voraus. Die Leistungseinheit „Identifikation und Authentisierung“ stellt der Leistungseinheit „Zugriffskontrolle“ alle zur Verwaltung und Kontrolle von Zugriffsrechten erforderlichen Identifikations- und Authentisierungsdaten zur Verfügung.

Dazu zählen u. a. Informationen über die Ermächtigung und Berechtigung einzelner Nutzer.

LSIC03.S.5 Schnittstelle zu LSIC04 „Auditing“

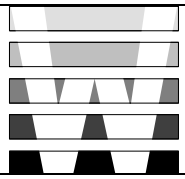
LSIC03.S.5.1 Übergabe von Protokollierungsinfomationen
Alle zur Protokollierung von Zugriffsversuchen erforderlichen Informationen werden von der Leistungseinheit „Zugriffskontrolle“ an die Leistungseinheit „Auditing“ übergeben.

LSIC03.S.5.2 Schutz der Audit-Mechanismen, -Daten und -Parameter
Die Leistungseinheit „Zugriffskontrolle“ gewährleistet, daß außer dem Auditverantwortlichen kein Nutzer auf die Audit-Mechanismen, -Parameter und Protokolldaten zugreifen kann.

LSIC03.S.5.3 Schutz der Nutzerprofile
Die Leistungseinheit „Zugriffskontrolle“ gewährleistet, daß außer dem Auditverantwortlichen kein Nutzer auf die Nutzerprofile zur Intrusion Detection zugreifen kann.

3.5.3.3.2 Anforderungen an die Methodenunterstützung

keine



3.5.3.3.3 Anforderungen an die Funktionen

LSIC03.F.1 Granularität der Rechteverwaltung

LSIC03.F.1.1 Subjekte
Es ist möglich, die Zugriffsrechte einzelner Nutzer oder definierter Nutzergruppen, von Rollen oder von Prozessen zu unterscheiden. Nutzergruppen und Rollen können (nur) von speziell autorisierten Nutzern definiert werden.

LSIC03.F.1.2 Objekte
Es ist möglich, die Zugriffsrechte auf einzelne Objekte oder definierte Typen von Objekten zu unterscheiden. Neue Typen von Objekten können nur von autorisierten Nutzern definiert werden.

LSIC03.F.1.3 Kapselung
Es ist möglich, den Zugriff auf bestimmte Objekte nur über festgelegte Prozesse zuzulassen.

LSIC03.F.1.4 Zugriffsarten
Mindestens folgende Zugriffsarten werden bei der Zugriffskontrolle unterschieden:
– Lesen von Objekten,
– Schreiben von Objekten,
– Anlegen von Objekten,
– Löschen von Objekten,
– Umbenennen von Objekten,
– Ausführen von ausführbaren Objekten.

LSIC03.F.2 Rechteverwaltung

LSIC03.F.2.1 Rechtevergabe
Es besteht die Möglichkeit, entsprechend der obigen Granularität Zugriffsrechte für Objekte in Form einer beliebigen Kombination der möglichen Zugriffsarten an Subjekte zu vergeben.

Insbesondere ist es auch möglich, einem Subjekt jeden Zugriff bezüglich eines Objekts zu verwehren.

LSIC03.F.2.2 Änderung und Zurücknahme von Rechten
Es ist möglich, bestehende Zugriffsrechte entsprechend der obigen Granularität zu verändern oder zurückzunehmen.

LSIC03.F.3 Schutz der Rechteverwaltungsdaten
Es ist sichergestellt, daß nur dem Systemverwalter schreibender Zugriff auf alle Daten der Rechteverwaltung gewährt wird.

LSIC03.F.4 Bedingungen
Es ist möglich, an die Zugriffsrechte zusätzliche Bedingungen zu knüpfen.

Z. B.: Der Zugriff ist nur an bestimmten Tagen und zu bestimmten Uhrzeiten gestattet.

LSIC03.F.5 Rollen
Die Zugriffsrechte können zur Unterstützung von Rollen zusammengefaßt werden. U. a. sind die Rollen Operator, Systemverwalter und Sicherheitsbeauftragter definierbar.

Es ist auch möglich, eine Rolle „Datenadministrator“ zu definieren. Dies ist die einzige Rolle, die Schemaänderungen vornehmen kann. Außerdem ist es möglich, Nutzer in Nutzergruppen zu organisieren und Rechte und Voreinstellungen für einen Nutzer oder eine ganze Nutzergruppe in jeweils einem Profil festzulegen.

LSIC03.F.6 Positivliste
Es ist möglich, zu einem beliebigen Objekt eine Liste von zugriffsberechtigten Nutzern sowie Nutzergruppen zusammen mit ihren jeweiligen Zugriffsrechten für dieses Objekt zu erstellen.

LSIC03.F.7 Negativliste
Es ist möglich, zu einem beliebigen Objekt eine Liste derjenigen Nutzer sowie Nutzergruppen zu erstellen, denen bestimmte Zugriffe auf dieses Objekt verweigert sind.

Z. B. könnte diese Liste alle Nutzer angeben, die für ein bestimmtes Objekt entweder kein Schreibrecht oder überhaupt kein Zugriffsrecht haben.

LSIC03.F.8 Regelbasierte Zugriffskontrolle (Mandatory Access Control)

LSIC03.F.8.1 Existenz einer regelbasierten Zugriffskontrolle
Die regelbasierte Zugriffskontrolle erfolgt nach festgelegten, unveränderbaren, eindeutigen Regeln. Als Entscheidungsgrundlage für die Prüfung dieser Regeln dienen spezielle Attribute (z. B. Einstufungen), mit denen alle Subjekte und alle der Objektverwaltung unterliegenden Objekte versehen sind. Die Regeln legen fest, welche Kombinationen der Attributwerte von Subjekt und Objekt für ein Subjekt nötig sind, um Zugriff auf dieses Objekt zu erhalten. Die Vergabe von Zugriffsrechten erfolgt durch Wertzuweisung an die Attribute. Beim Export eines Objektes werden seine Attribute so mitexportiert, daß der Empfänger diese eindeutig rekonstruieren kann.

LSIC03.F.8.2 Attributvergabe
Ein speziell autorisierter Nutzer ist in der Lage, beim Import nicht-attributierter Daten diesen Attribute (z. B. Einstufungen) zuzuordnen.

LSIC03.F.8.3 Ausgestaltung
Die regelbasierten Zugriffsrechte sind so gestaltet, daß folgender Sonderfall realisiert werden kann:
Das Attribut besteht aus zwei Teilen. Der erste Teil besitzt hierarchisch geordnete Werte, der zweite stellt eine Menge dar. Ein Subjekt kann nur Objekte
1. lesen, deren Attribut von dem des Subjekts dominiert wird,
2. schreiben, deren Attribut das des Subjekts dominiert.

Im Amtsbereich enthält der erste Teil Einstufungen, z. B. offen, VS-vertraulich, geheim. Der zweite Teil enthält Kategorien. Ein Attribut A dominiert ein Attribut B genau dann, wenn: Teil 1 von A hierarchisch größer oder gleich Teil 1 von B ist und Teil 2 von B eine Teilmenge von Teil 2 von A ist.

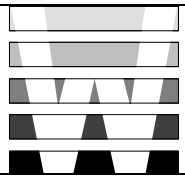
LSIC03.F.8.4 Anzeige der Attributwerte
Jeder Nutzer hat die Möglichkeit, sich die Attributwerte aller Prozesse, die für ihn während einer laufenden interaktiven Sitzung aktiv sind, anzeigen zu lassen.

LSIC03.F.8.5 Übertragung

LSIC03.F.8.5.1 Einstufungsmöglichkeiten von Kommunikationskanälen
Jeder Kommunikationskanal ist als einstufig oder mehrstufig identifizierbar. Dadurch ist definierbar, welche Einstufung eine Nachricht besitzen muß, damit sie über diesen Kommunikationskanal transportiert werden darf.

Z. B. dürfen über einstufige Kanäle nur Nachrichten mit einer bestimmter Einstufung gesendet werden. Alle anderen Nachrichten werden von diesem Kanal nicht weitergeleitet.

LSIC03.F.8.5.2 Einstufige Kanäle
Als einstufig gekennzeichnete Kanäle sind mit einem Attribut versehen, das ihre Einstufung wiedergibt. Über sie dürfen nur Daten gesendet oder empfangen werden, wenn diese mit dem entsprechenden Attribut versehen sind und der Wert mit dem vorab definierten Attributwert des Kanals übereinstimmt.



LSIC03.F.8.5.3 Mehrstufige Kanäle
Es ist möglich, eine minimale und maximale Einstufung für den Kanal anzugeben. Daten werden von dem Kanal nur transportiert, wenn sich deren Einstufung zwischen der minimalen und maximalen Einstufung bewegt. Bei der Ausgabe eines Objekts an ein mehrstufiges Ein-/Ausgabegerät wird die dem Objekt zugewiesene Einstufung ebenfalls ausgegeben.

LSIC03.F.8.5.4 Verwaltung der Kanaleinstufungen
Es besteht die Möglichkeit, Kanaleinstufungen zu vergeben, zu verändern und zurückzunehmen. Jeder dieser Vorgänge ist protokollierbar.

LSIC03.F.8.5.5 Änderung der Einstufung der Kanäle
Nur autorisierte Nutzer können die Einstufung der Kanäle ändern.

LSIC03.F.8.5.6 Übertragung von Daten und ihren Attributen
Das Übertragungsprotokoll stellt sicher, daß auf der Empfängerseite alle Daten und Attribute vollständig und eindeutig rekonstruierbar sind und dabei die Attribute den Daten eindeutig zugeordnet werden können.

LSIC03.F.8.5.7 Menschenlesbare Ausgaben
Menschenlesbare Ausgaben werden immer mit einem die Einstufung kennzeichnenden Attribut versehen.

Menschenlesbare Ausgaben sind z. B. Drucker- oder Bildschirmausgaben.

LSIC03.F.9 Nutzerbestimmbare Zugriffskontrolle (Discretionary Access Control)

LSIC03.F.9.1 Existenz einer benutzerbestimmbaren Zugriffskontrolle
Jeder Nutzer bzw. jeder in seinem Auftrag handelnde Prozeß kann bestimmen, welche Zugriffsrechte andere Nutzer bzw. deren Prozesse bezüglich den seiner Kontrolle unterliegenden Objekten besitzen sollen.

Die Kontrolle über ein Objekt hat im allgemeinen der Eigentümer des Objekts sowie der Systemverwalter.

LSIC03.F.9.2 Änderung der Zugriffsrechte
Der Eigentümer eines Objekts kann die Zugriffsrechte jederzeit ändern.

Diese Rechte sollte er gemäß dem Need-to-know-Prinzip vergeben.

LSIC03.F.9.3 Kontrolle vergeben
Für speziell autorisierte Nutzer (z. B. Systemverwalter) besteht die Möglichkeit, für jedes Objekt einen bestimmten Nutzer oder eine Gruppe von Nutzern festzulegen, der/die die Kontrolle über das Objekt übernehmen soll.

LSIC03.F.10 Zugriffsrechteprüfung
Bei jedem Zugriff eines lokalen oder entfernten Subjekts auf ein der Objektverwaltung unterliegendes Objekt wird die Berechtigung des Subjekts überprüft. Insbesondere unterliegt die Weitergabe von Zugriffsrechten, das Einbringen neuer Nutzer, das Löschen von Nutzern und das zeitweilige Löschen aller Rechte eines Nutzers der Rechteverwaltung. Falls sowohl die nutzerbestimmbare als auch die regelbasierte Zugriffskontrolle im System implementiert sind, gilt ein Zugriff dann als berechtigt, wenn beide Kontrollen erfolgreich durchlaufen wurden. Unberechtigte Zugriffsversuche werden abgewiesen.

LSIC03.F.11 Empfangene Objekte
Empfangene Objekte werden unverzüglich der Zugriffskontrolle übergeben.

LSIC03.F.12 Default-Rechte
Es ist konfigurierbar, daß bei der Einrichtung neuer Nutzer oder bei der Erzeugung neuer Objekte diese mit vordefinierten Zugriffsprofilen versehen werden.

LSIC03.F.13 Schutz vor Abhören des Verbindungsaufbaus
Der Zugriff auf die niedrigeren Netzwerk-Protokolldienste ist für unprivilegierte Nutzer gesperrt.

LSIC03.F.14 Selektiver Filter
Bei dieser Leistungseinheit werden auf der Transport-Ebene die einzelnen übertragenen Pakete in Abhängigkeit von den Adressen der beteiligten Kommunikationspartner und des gewählten Dienstes gefiltert. Über Kontrollisten wird auf dem Zielrechner geprüft, welcher Rechner Zugang zu welchen Dienstleistungen des Zielrechners erhält. Somit kann der Zugang zu Diensten (wie z. B. UNIX-Dienste Telnet, Remote-Login, ftp, usw.) unterbunden werden.

LSIC03.F.15 Sicheres Weiterleiten
Neben den Adressen werden auch die Port-Nummern ausgewertet, die bestimmen, zu welchem Dienst die Nachrichten/Nachrichten-Pakete gehören. Falls in der Zugriffskontrolliste der Zugang von Nachrichten/Nachrichten-Paketen dieser Quell-Adresse zu dieser Ziel-Adresse mit diesem Ziel-Port gestattet ist, so wird das Paket entweder weitervermittelt, wenn der „intelligente Router“ nur eine Zwischenstation ist, oder es wird an die nächste Protokollschicht gereicht und dort weiterverarbeitet. Findet sich in der Zugriffskontrolliste kein Eintrag oder ein negativer Eintrag, so wird das Paket i. a. nicht an die höheren Protokollschichten weitergeleitet.

3.5.3.3.4 Sonstige Anforderungen

keine

3.5.4 Leistungseinheit LSIC04 „Auditing“

3.5.4.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.5.4.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen zusammengefaßt, die sicherstellen, daß über Handlungen, die von Nutzern bzw. von Prozessen im Namen solcher Nutzer ausgeführt werden, Informationen aufgezeichnet werden, damit die Folgen solcher Handlungen später dem betreffenden Nutzer zugeordnet werden können und der Nutzer für seine Handlungen verantwortlich gemacht werden kann. Im Systemverwalterhandbuch sollte der Auditverantwortliche darauf hingewiesen werden, daß die Aufzeichnung von Nutzerverhalten in Übereinstimmung mit den relevanten gesetzlichen Regelungen (u. a. Bundes- und Landesdatenschutzgesetze) zu erfolgen hat.

Darin eingeschlossen sind Nachprüfungen, durch die später festgestellt werden kann, ob tatsächlich Sicherheitsverletzungen vorgelegen haben und welche Informationen oder sonstige Betriebsmittel davon betroffen waren.

Diese Leistungseinheit umfaßt alle Funktionen, die

- zur Aufzeichnung sicherheitsrelevanter Ereignisse (erfolgte und versuchte Ausübung von Rechten),
- zur Auffindung und Überprüfung von Ereignissen, die eine Bedrohung für die Sicherheit darstellen können und
- zur Ergreifung von Gegenmaßnahmen

vorgesehen sind.

3.5.4.3 Anforderungen

3.5.4.3.1 Anforderungen an die Schnittstellen

LSIC04.S.1 Schnittstelle zu LSIC02 „Identifikation und Authentisierung“
Der Mechanismus zur Identifizierung und Authentisierung von Nutzern erlaubt es, berechtigte Nutzer eindeutig gegenüber der Leistungseinheit „Auditing“ zu identifizieren. Bei jeder Authentisierung, ob erfolgreich oder erfolglos, werden die für die Protokollierung dieses Ereignisses notwendigen Informationen (u. a. Parameter, Erfolg bzw. Mißerfolg des Ereignisses) an die Leistungseinheit „Auditing“ übergeben.

LSIC04.S.2 Schnittstelle zu LSIC03 „Zugriffskontrolle“

LSIC04.S.2.1 Übergabe von Protokollierungsinfomationen
Alle zur Protokollierung von Zugriffsversuchen erforderlichen Informationen werden von der Leistungseinheit „Zugriffskontrolle“ an die Leistungseinheit „Auditing“ übergeben.

LSIC04.S.2.2 Schutz der Audit-Mechanismen, -Daten und -Parameter
Die Leistungseinheit „Zugriffskontrolle“ gewährleistet, daß außer dem Auditverantwortlichen kein Nutzer auf die Audit-Mechanismen, -Parameter und Protokolldaten zugreifen kann.

Insbesondere ist der Zugriff auf die im Speicher, z. B. in Puffern, befindlichen Protokolldaten ausschließlich über die Leistungseinheit LSIC04 „Auditing“ und nur dem Auditverantwortlichen möglich.

LSIC04.S.2.3 Schutz der Nutzerprofile
Die Leistungseinheit „Zugriffskontrolle“ gewährleistet, daß außer dem Auditverantwortlichen kein Nutzer auf die Nutzerprofile zur Intrusion Detection zugreifen kann.

Damit ist der Datenschutz gewährleistet.

LSIC04.S.3 Schnittstelle zu vertrauenswürdigen Komponenten
Es existiert eine Schnittstelle zu vertrauenswürdigen Komponenten. Diese ermöglicht, daß vertrauenswürdige Komponenten ihre eigenen Audit-Records für den Audit-Trail schreiben. Um redundante Aufzeichnungen zu vermeiden, erhalten diese vertrauenswürdigen Komponenten gleichzeitig das Privileg, die Protokollierung ihrer Aktivitäten durch die Audit-Komponente auszuschalten.

Diese Schnittstelle gestattet die Aufzeichnung wesentlich aussagekräftigerer Audit-Daten als es der Leistungseinheit „Auditing“ selbst möglich ist.

3.5.4.3.2 Anforderungen an die Methodenunterstützung

keine

3.5.4.3.3 Anforderungen an die Funktionen

LSIC04.F.1 Auditadministrator
Die Rolle eines Auditadministrators ist definierbar. Sie zeichnet sich durch spezielle Privilegien bezüglich der Beweissicherung aus.

Spezielle Privilegien sind z. B. Anstoßen, Ausschalten der Protokollierung oder Zugriff auf die Mechanismen, Parameter und Protokolldaten.

LSIC04.F.2 Erzeugen und Verwalten des Audit-Trails

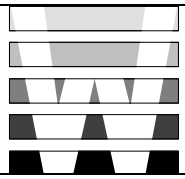
LSIC04.F.2.1 Aufzeichnungsmechanismus
Die Leistungseinheit „Auditing“ stellt einen Aufzeichnungsmechanismus für sicherheitsrelevante Ereignisse zur Verfügung. Dieser Mechanismus muß in der Lage sein, für alle sicherheitsrelevanten Ereignisse (siehe unter LSIC04.F) die erforderlichen Daten zu erzeugen, zu speichern, zu verwalten und für eine Auswertung zur Verfügung zu stellen.

LSIC04.F.2.2 Ein- bzw. Ausschalten
Es existiert ein Mechanismus zum Ein- und Ausschalten der Protokollierung.

LSIC04.F.2.3 Puffer
Der Audit-Trail wird in einem Puffer (im Kernelmemory) gespeichert. Es besteht die Möglichkeit, das Schreiben des Pufferinhalts auf ein nichtflüchtiges Medium anzustoßen.

LSIC04.F.2.4 Verwaltung
Es existieren vertrauenswürdige Mechanismen zur Verwaltung der Protokolldaten.

Dazu zählen z. B. Mechanismen zur Initialisierung, Modifikation der Steuerungsparameter, Datenkomprimierung und -expandierung.



LSIC04.F.3 Auswahl bestimmter Ereignisse

LSIC04.F.3.1 Voreinstellung
Gemäß der Voreinstellung werden alle protokollierbaren Ereignisse in den Audit-Trail aufgenommen.

LSIC04.F.3.2 Vorab-Auswahl

LSIC04.F.3.2.1 Positivauswahl
Es existiert ein Mechanismus zur Auswahl der Ereignisse, die in den Audit-Trail aufgenommen werden sollen. Es können dabei aber nicht beliebige Ereignisse von der Protokollierung ausgenommen werden. Z. B. werden grundsätzlich alle Aktionen privilegierter Nutzer vollständig aufgenommen.

Dadurch wird u. a. verhindert, daß der Auditverantwortliche sich selbst von der Protokollierung ausschließt.

LSIC04.F.3.2.2 Negativauswahl
Der Mechanismus ermöglicht auch die Angabe von Ereignissen, die explizit nicht aufgenommen werden sollen.

LSIC04.F.3.3 Nachträgliche Auswahl
Es existiert ein Mechanismus zur nachträglichen Auswahl.

Damit kann der Auditverantwortliche aus dem Audit-Trail diejenigen Ereignisse auswählen, die er analysieren will.

LSIC04.F.3.4 Granularität
Es ist möglich, Ereignisse auf der Basis von Nutzer-Identität, Nutzergruppen, Prozessen und Objekten auszuwählen. Nutzer und Objekte können auch aufgrund bestimmter Attributwerte (z. B. ab einer Einstufung höher als VS-vertraulich) ausgewählt werden.

LSIC04.F.3.5 Grundeinstellung
Es existiert eine Grundeinstellung für neue Nutzer.

Sie stellt sicher, daß zumindest die wichtigsten, durch einen neuen Nutzer initiierten Ereignisse protokolliert werden bis zu dem Zeitpunkt, zu dem der Auditverantwortliche den neuen Nutzer individuell in die Audit-Steuerung aufnimmt.

LSIC04.F.4 Protokollierbare Ereignisse

LSIC04.F.4.1 Identifikation und Authentisierung
Es ist möglich, die Benutzung der Identifizierungs- und Authentisierungsmechanismen zu protokollieren.

Dazu zählen u. a. die Prozeduren zum lokalen oder entfernten Login und zur Paßwortänderung.

LSIC04.F.4.2 Objektzugriff

LSIC04.F.4.2.1 Neuanlegen und Löschen von Objekten
Es ist möglich, das Neuanlegen und Löschen eines der Rechteverwaltung unterliegenden Objekts zu protokollieren.

LSIC04.F.4.2.2 Einbringen und Entfernen von Objekten
Es ist möglich, das Einbringen von Objekten in den Adreßraum eines Nutzers (z. B. Öffnen von Dateien oder Laden von Programmen) sowie das Entfernen von Objekten aus dem Adreßraum eines Nutzers (z. B. Schließen von Dateien oder Beenden einer Sitzung) zu protokollieren.

LSIC04.F.4.3 Einbringen und Entfernen von Datenträgern
Es ist möglich, das Einbringen oder Entfernen von Datenträgern zu protokollieren.

LSIC04.F.4.4 Rollen mit besonderen Rechten
Es ist möglich, das Einnehmen von Rollen mit besonderen Rechten zu protokollieren.

LSIC04.F.4.5 Nutzung von Privilegien
Es ist möglich, Aktionen von Personen mit besonderen Rechten, z. B. Auditadministrator oder Systemverwalter, zu protokollieren.

LSIC04.F.4.6 Änderungen sicherheitsrelevanter Systemdateien
Es ist möglich, alle Änderungen sicherheitsrelevanter Systemdateien zu protokollieren.

Dazu zählen z. B. das Anlegen, Sperren oder Löschen eines Nutzereintrags.

LSIC04.F.4.7 Attributierung
Es ist möglich, das Initialisieren und Ändern der Attribute von Subjekten und Objekten zu protokollieren.

Beispiele für solche Attribute sind u. a. Zugriffsrechte von Subjekten auf Objekte, Einstufungen von Nutzern, Prozessen, Datenobjekten oder Exportkanälen.

LSIC04.F.4.8 Typverwaltung
Es ist möglich, das Anlegen und Löschen von Objekttypen und die Vergabe eines Typs für ein Objekt zu protokollieren.

LSIC04.F.4.9 Systemwiederherstellung
Es ist möglich, alle Aktionen zur Systemwiederherstellung nach einem Systemabsturz zu protokollieren.

LSIC04.F.4.10 Starten und Stoppen des Systems
Es ist möglich, alle Aktionen zum Starten und Stoppen des Systems zu protokollieren.

LSIC04.F.4.11 Erzeugen von Druckausgaben
Es ist möglich, das Erzeugen von Druckausgaben zu protokollieren.

LSIC04.F.4.12 Verdeckte Kanäle
Es ist möglich, bekannte Ereignisse zu protokollieren, die ausgenützt werden können, um einen nicht autorisierten Informationsfluß durch Ausnützung eines verdeckten Kanals zu erzeugen.

LSIC04.F.4.13 Datenübertragung
Es ist möglich, den Verbindungsaufbau zur Datenübertragung, erkannte Fehler bei der Datenübertragung, Datenverluste und -veränderungen beim Datenimport sowie spezielle Datenübertragungsaktionen zu protokollieren.

LSIC04.F.5 Erforderliche Daten pro Ereignis

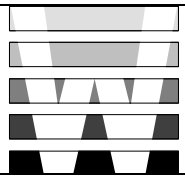
LSIC04.F.5.1 Datum und Uhrzeit
Es werden für jedes Ereignis Datum und Uhrzeit seines Eintretens in den Audit-Trail aufgenommen.

LSIC04.F.5.2 Nutzeridentität
Es wird für jedes Ereignis die eindeutige Identifizierung des Nutzers, der das Ereignis angestoßen hat, in den Audit-Trail aufgenommen.

LSIC04.F.5.3 Art des Ereignisses
Es wird für jedes Ereignis die Art des Ereignisses in den Audit-Trail aufgenommen.

LSIC04.F.5.4 Erfolg/Mißerfolg
Es wird für jedes Ereignis protokolliert, ob es erfolgreich war oder nicht.

LSIC04.F.5.5 Herkunft
Es wird für jedes Ereignis die Herkunft des Ereignisses, z. B. Terminal ID oder Netzadresse, in den Audit-Trail aufgenommen.



LSIC04.F.5.6 Objektname
Bei Ereignissen, die Objekte betreffen, wird der Name des Objekts aufgezeichnet.

LSIC04.F.5.7 Attribute
Es werden sämtliche Attribute der an dem Ereignis beteiligten Subjekte und Objekte aufgezeichnet.

Bei der Datenübertragung sind der Sender und der Empfänger die betreffenden Subjekte und die Nutzdaten entsprechen dem Objekt.

LSIC04.F.5.8 Identifizierung und Authentisierung
Bei der Benutzung der Identifizierungs- und Authentisierungsmechanismen wird der Name des zu identifizierenden Subjekts aufgezeichnet.

LSIC04.F.5.9 Paßwortschutz
Bei Eingabe eines falschen Namens während einer Identifikations-/Authentisierungsprozedur (z. B. Login-Prozedur) kann die Aufzeichnung des Namens unterbunden werden, um sicherzustellen, daß bei versehentlicher Eingabe des Paßwortes an falscher Stelle dieses nicht kompromittiert wird.

Es ist im konkreten Fall abzuwägen, ob trotz der Gefahr der Paßwortkompromittierung der Kennungsname zugunsten einer nachträglichen Aufklärung aufgezeichnet werden soll.

LSIC04.F.5.10 Änderungsbeschreibungen
Es wird eine Beschreibung der Änderungen, die der Systemverwalter an den für Nutzerverwaltung und Systemsicherheit relevanten Dateien vornimmt, aufgezeichnet.

LSIC04.F.5.11 Verbindungsaufbau
Beim Verbindungsaufbau für eine Datenübertragung werden der Name des Empfängers (Rechner, Prozeß oder Nutzer) und die Parameter des Verbindungsaufbaus protokolliert.

LSIC04.F.5.12 Erkannte Fehler bei der Datenübertragung
Werden bei der Datenübertragung Fehler erkannt, so werden die Art des Fehlers, der Erfolg oder Mißerfolg des Korrekturversuchs und die Nutzer-IDs der Kommunikationspartner aufgezeichnet.

LSIC04.F.5.13 Datenübertragung
Es ist möglich bei Datenübertragung die Nutzer-IDs von Sender und Empfänger, die übertragenen Nutzinformationen sowie Datum und Uhrzeit des Empfangs der Daten aufzuzeichnen.

LSIC04.F.6 Besonders kritische Ereignisse

LSIC04.F.6.1 Systemverhalten bei besonders kritischen Ereignissen
Es besteht die Möglichkeit zur Überwachung von Ereignissen, die besonders sicherheitsrelevant sind oder die aufgrund der Häufigkeit ihres Auftretens eine kritische Bedrohung der Sicherheit der SEU darstellen können. Beim Auftreten solcher Ereignisse wird der Auditverantwortliche unverzüglich informiert, und es werden Maßnahmen in die Wege geleitet, durch welche ein weiteres Auftreten solcher Ereignisse unterbunden wird.

LSIC04.F.6.2 Überlauf

LSIC04.F.6.2.1 Erkennung
Es ist möglich zu erkennen, wenn das zur Aufnahme des Audit-Trails ausgewählte Speichermedium voll ist. Die daraufhin eingeleitete Aktion wird protokolliert.

LSIC04.F.6.2.2 Warnung
Sobald die Speicherkapazitäten, die für den Audit-Trail vorgesehen sind, erschöpft sind, wird der Auditverantwortliche benachrichtigt.

LSIC04.F.6.2.3 Retten der Protokolldaten
Auch im Falle eines Speicherüberlaufs werden alle Daten des Audit-Trails archiviert.

LSIC04.F.7 Verfügbarkeit der Audit-Daten
Die Audit-Daten stehen innerhalb von Minuten nach ihrer Aufzeichnung dem Auditverantwortlichen zur Verfügung.

LSIC04.F.8 Vertrauenswürdige Auswertemechanismen

LSIC04.F.8.1 Intrusion Detection
Es existieren vertrauenswürdige Auswertemechanismen, u. a. zur statistischen Auswertung und gezielten Datenabfrage über verschiedene Ereignisparameter sowie zur Ausgabe der betreffenden Daten. Sie ermöglichen es, unabhängig von der zu überwachenden SEU die Audit-Daten online gezielt auf Eindringversuche hin zu analysieren. Dazu wird das Verhalten von Subjekten auf Abweichungen von ihrem „Normalverhalten“ hin untersucht.

LSIC04.F.8.2 Subjekte
Es ist möglich, Subjekte als individuelle Nutzer, Gruppen von Nutzern, Remote Hosts und als das ganze System zu definieren.

Der Vorteil, eine ganze Gruppe als ein Subjekt zu betrachten, liegt darin, daß Abweichungen festgestellt werden können, die nicht einem einzelnen Gruppenmitglied zugeordnet werden können (z. B. Anstieg der vergeblichen Login-Versuche unter verschiedenen Nutzerkennungen).

LSIC04.F.8.3 Nutzerprofil
Es ist möglich, für jedes definierte Subjekt ein Verhaltensprofil in einer Datenbasis abzuspeichern. Das Profil beschreibt das normale (d. h. das erwartete) Verhalten eines Subjekts bzgl. einer Menge von Parametern, welche Hinweise auf mögliche Sicherheitsverletzungen geben.

Die Parameter sind beispielsweise

– für individuelle Nutzer:
  - *Verbrauchte CPU-Zeit,*
  - *Anzahl der I/O-Aktivitäten,*
  - *Anzahl der erzeugten Audit-Records,*
  - *Zugriffe auf Verzeichnisse,*
  - *zu welchen Zeiten (an welchen Tagen, zu welchen Stunden, ...) gearbeitet wird,*
  - *von welchen Geräten aus der Zugriff erfolgt;*

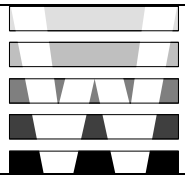
– für Remote Hosts:
  - *Anzahl der fehlgeschlagenen Login-Versuche,*
  - *Art der Netzaktivitäten;*

– für überwachte Systeme:
  - *Anzahl der fehlgeschlagenen Login-Versuche,*
  - *Anzahl der Systemfehler.*

Die Profile werden laufend an das aktuelle Nutzerverhalten angepaßt, indem neugemessene Parameterwerte eingearbeitet werden. Erfolglose sowie erfolgreiche Eindringversuche eines Nutzers werden anhand signifikanter Abweichungen von seinem typischen Verhalten erkannt.

LSIC04.F.8.4 Sicherheitslücken beim Systemzugang
Bekannte Systemverwundbarkeiten, die ein unerlaubtes Eindringen ermöglichen, werden gezielt überwacht.

LSIC04.F.8.5 Sicherheitslücken beim Objektzugriff
Handlungsweisen, die es ermöglichen, die Rechteprüfung zu umgehen oder unerlaubt zu Rechten zu gelangen, werden gezielt überwacht.



LSIC04.F.9 Auskunft
Es steht ein Mechanismus zur Verfügung, der es erlaubt, für jedes protokollierbare Ereignis festzustellen, ob es in den Audit-Trail aufgenommen werden muß.

3.5.4.3.4 Sonstige Anforderungen

keine

3.5.5 Leistungseinheit LSIC05 „Wiederaufbereitung“

3.5.5.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.5.5.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen zusammengefaßt, die sicherstellen, daß Betriebsmittel wie beispielsweise Hauptspeicher oder Bereiche von Plattenspeichern vor einer Wiederverwendung durch andere Nutzer so aufbereitet werden, daß keine Rückschlüsse auf ihren früheren Inhalt möglich sind.

Darin eingeschlossen sind alle Funktionen, die für die Überwachung der Wiederaufbereitung von Objekten vorgesehen sind.

Die Leistungseinheit umfaßt alle Funktionen, die

- zur Initialisierung wiederverwendbarer Medien (z. B. Disketten),
- zur Löschung der Daten auf wiederverwendbaren Medien,
- zur Löschung der Ausgaben auf Ausgabegeräten, wenn diese nicht in Betrieb sind (z. B. Bildschirm)

vorgesehen sind.

3.5.5.3 Anforderungen

3.5.5.3.1 Anforderungen an die Schnittstellen

LSIC05.S.1 Schnittstelle zur Leistungseinheit LSIC06 „Unverfälschtheit, Vertraulichkeit und Verbindlichkeit“
Die Leistungseinheit LSIC06 kann die Wiederaufbereitung wiederverwendbarer Speichermedien veranlassen.

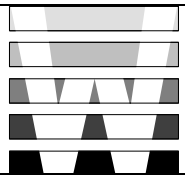
3.5.5.3.2 Anforderungen an die Methodenunterstützung

keine

3.5.5.3.3 Anforderungen an die Funktionen

LSIC05.F.1 Granularität
Als Speicherobjekte, die einer Wiederaufbereitung unterzogen werden, unterscheidet die SEU mindestens zwischen

- CPU-Register,
- Cachespeicher,
- Fließpunkt-Coprozessor-Register,
- Speicher von I/O-Controllern,
- Hauptspeicher,
- Blöcke oder Sektoren von Speicherplatten (magnetisch oder optisch),
- Blöcke oder Sektoren von Magnetbändern,
- Disketten.



LSIC05.F.2 Wiederaufbereitung

LSIC05.F.2.1 Mechanismus
Alle Speicherobjekte werden vor einer Wiederverwendung durch andere Nutzer oder Prozesse so aufbereitet, daß keine Rückschlüsse auf ihren früheren Inhalt möglich sind.

LSIC05.F.2.2 Ausprägungen

LSIC05.F.2.2.1 Löschen
Für jedes Speicherobjekt wird entweder bei Allokation oder bei Deallokation sein bisheriger Inhalt gelöscht.

LSIC05.F.2.2.2 Überschreiben
Für jedes Speicherobjekt wird entweder bei Allokation oder bei Deallokation sein bisheriger Inhalt mit neuen Daten überschrieben, auf die ein neues Subjekt Zugriff haben darf (Initialisierung).

LSIC05.F.3 Explizite Wiederaufbereitung
Es ist möglich, die physikalische Löschung von Informationen in Speicherobjekten explizit auszulösen.

LSIC05.F.4 Automatische Wiederaufbereitung
Sobald ein Speicherobjekt nach erfolgtem Zugriff wieder freigegeben werden soll, wird sein Inhalt automatisch gelöscht oder gezielt überschrieben.

LSIC05.F.5 Wiederaufbereitung bei Wiederanlauf
Bei einem Wiederanlauf werden alle Speicherobjekte, die nicht wiederaufbereitet worden sind, automatisch einer Wiederaufbereitung unterzogen.

Daß Speicherobjekte nicht wiederaufbereitet wurden, kann z. B. durch einen Systemausfall verursacht worden sein.

3.5.5.3.4 Sonstige Anforderungen

keine

3.5.6 Leistungseinheit LSIC06 „Unverfälschtheit, Vertraulichkeit und Verbindlichkeit“

3.5.6.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.5.6.2 Kurzcharakteristik

Die in dieser Leistungseinheit zusammengefaßten Anforderungen stellen zum einen sicher, daß bestimmte Beziehungen zwischen unterschiedlichen Daten erhalten bleiben (Konsistenz), daß Nachrichten (Daten, die zwischen verschiedenen Rechnern ausgetauscht werden) unverändert übertragen werden und daß Manipulationen an Daten, Nachrichten oder Programmen erkannt werden. Zum anderen gewährleisten diese Anforderungen die Vertraulichkeit und die Verbindlichkeit von Daten.

Die Leistungseinheit umfaßt alle Funktionen, die

- zur Prüfung und Sicherstellung und Authentizität von Nachrichten,
- zur Überprüfung und Sicherstellung der Unverfälschtheit von Daten, Nachrichten oder Progammen,
- zur Aufrechterhaltung der Unverfälschtheit von Daten, Nachrichten (inklusive ihrer Absender- und Empfängeradressen) oder Programmen,
- zur Entdeckung von Verlusten, Ergänzungen, Veränderungen von Daten, Nachrichten oder Programmen,
- zur Sicherstellung der Vertraulichkeit von Daten oder Nachrichten,
- zur Sicherstellung der Verbindlichkeit von Nachrichten

vorgesehen sind.

3.5.6.3 Anforderungen

3.5.6.3.1 Anforderungen an die Schnittstellen

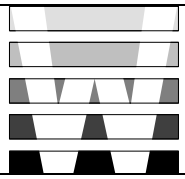
LSIC06.S.1 Schnittstelle zu LSIC02 „Identifikation und Authentisierung“
Die Authentifikationsdaten werden nur verschlüsselt gespeichert oder übertragen. Die Leistungseinheit „Unverfälschtheit, Vertraulichkeit und Verbindlichkeit“ setzt dabei ein sicheres Einweg-Verfahren ein.

Dadurch werden gezielte Fälschungen von Authentisierungsdaten verhindert. Es ist gewährleistet, daß generell kein Einblick in die Authentisierungsdaten (z. B. Paßwörter) erfolgen kann – auch nicht durch den Systemverwalter. U. a. wird auch verhindert, daß ein Eindringling durch Abhören einer Verbindungsleitung (z. B. zwischen Terminal und Rechner) oder durch lesenden Zugriff auf die Paßwortdatei in den Besitz eines unverschlüsselten Paßwortes gelangt.

LSIC06.S.2 Schnittstelle zu LSIC05 „Wiederaufbereitung“
Die Leistungseinheit „Unverfälschtheit, Vertraulichkeit und Verbindlichkeit“ kann die Wiederaufbereitung wiederverwendbarer Speichermedien veranlassen.

3.5.6.3.2 Anforderungen an die Methodenunterstützung

keine



3.5.6.3.3 Anforderungen an die Funktionen

LSIC06.F.1 Plausibilitätskontrollen
Um den Nutzer vor fehlerhaften Dateneingaben und Datenänderungen zu schützen, werden Format-, Plausibilitäts- und Konsistenzprüfungen durchgeführt.

Die Prüfungen müssen in Abhängigkeit vom Typ einer Datei definiert werden.

LSIC06.F.2 Bestätigen von sicherheitskritischen Aktionen
Sensitive Aktionen sind vor der tatsächlichen Durchführung vom Nutzer nochmals zu bestätigen.

Dies soll verhindern, daß eine eventuell sicherheitsgefährdende Aktion (wie z. B. Löschen von Dateien) versehentlich angestoßen wird. Eine solche Aktion kann hiermit noch vor der Ausführung gestoppt werden.

LSIC06.F.3 Authentizität von Daten, Nachrichten und Programmen

LSIC06.F.3.1 Vertraulichkeit

LSIC06.F.3.1.1 Verschlüsselung zur Sicherstellung der Vertraulichkeit von Daten
Die Vertraulichkeit von Daten wird durch (symmetrische oder asymmetrische) Verschlüsselung sichergestellt.

Besonders zu schützende Daten sollten nur verschlüsselt gespeichert oder gesichert werden. Für weniger sensitive Daten ist der Schutz durch die Zugangs- und Zugriffskontrolle ausreichend. Die Beurteilung, welche Daten besonders schützenswert sind und welche nicht, trifft im konkreten Fall der Auftraggeber bzw. der Entwickler. Zugleich wird durch diese Verschlüsselung auch eine gezielte Manipulation der Daten erschwert, da ein Angreifer für eine gezielte Manipulation oft einen Zugang zu den unverschlüsselten Daten benötigt.

LSIC06.F.3.1.2 Verschlüsselung zur Sicherstellung der Vertraulichkeit von Nachrichten
Die Vertraulichkeit von Nachrichten wird durch (symmetrische oder asymmetrische) Verschlüsselung sichergestellt.

Sensitive Nachrichten sollten nur verschlüsselt versandt werden. Soll der Nachrichtenaustausch komplett verschlüsselt erfolgen, so ist eine End-zu-End-Verschlüsselung notwendig, die dann der Netzwerkschnittstelle vorgeschaltet ist. Jeder Rechner, der für den Transport solcher Nachrichten benötigt wird, muß mit der entsprechenden Verschlüsselungs-Soft- und -Hardware ausgestattet sein.
Dieser Mechanismus soll eine Geheimhaltung der Nutzdaten und der Empfängerinformationen über weite Teile des Übertragungsweges gewährleisten. Zusätzlich muß die Geheimhaltung des Übertragungsaufkommens auf speziellen Datenübertragungskanälen garantiert werden.
Zugleich wird durch diese Verschlüsselung auch eine gezielte Manipulation der Nachrichten erschwert, da bei geeigneter Wahl des Verschlüsselungsalgorithmus eine gezielte Manipulation nur an unverschlüsselten Daten vorgenommen werden kann.

LSIC06.F.3.2 Integrität

LSIC06.F.3.2.1 Integritätsbedingungen
Es ist möglich, Integritätsbedingungen zu formulieren.

LSIC06.F.3.2.2 Integritätsverletzung
Die Verletzung von Integritätsbedingungen führt automatisch zur Ausführung von Maßnahmen, die vom Nutzer definiert werden können.

LSIC06.F.3.2.3 Integritätsprüfung
Es ist möglich, Integritätsprüfungen explizit anzustoßen.

LSIC06.F.3.2.4 Prozedursprache
Es ist möglich, Integritätsbedingungen unter dem Einsatz einer Prozedursprache festzulegen.

LSIC06.F.3.3 Nicht-kryptografische Prüfsummen

LSIC06.F.3.3.1 Integritätsschutz von Daten
Um die Integrität von Daten zu gewährleisten, werden Prüfsummen zu den Daten berechnet.

Wenn ein Angreifer die Daten manipuliert, so wird dies beim Vergleich von Original-Prüfsumme und aktueller Prüfsumme erkannt, sofern der Angreifer nicht die Originalprüfsumme anpassen kann oder die Manipulation so durchführt, daß die Prüfsumme gleich bleibt.

LSIC06.F.3.3.2 Integritätsschutz von Programmen
Es werden wie für die Daten Prüfsummen berechnet. Da sich Programme nicht ändern, können die Prüfsummen dieser Programme schreibgeschützt verwahrt werden.

LSIC06.F.3.3.3 Integritätsschutz von Nachrichten
Es werden wie für die Daten Prüfsummen berechnet. Mittels verschiedener Prüfsummen einer Nachricht können Übertragungsfehler in Nachrichten lokalisiert und behoben werden.

LSIC06.F.3.4 Kryptografische Prüfsummen

LSIC06.F.3.4.1 Integritätsschutz von Daten
Die Integrität der Daten wird mit Hilfe kryptografischer Prüfsummenbildung (elektronische Versiegelung mittels symmetrischer oder asymmetrischer Verschlüsselungsverfahren) sichergestellt. Die kryptografischen Prüfsummen werden hierbei bei Überprüfung der Daten durch denselben Verschlüsselungsalgorithmus (bei symmetrischem Verschlüsselungsverfahren mit demselben oder einem hieraus ableitbaren Schlüssel, bei asymmetrischen Verfahren mit dem öffentlichen Schlüssel) neu berechnet und mit der gespeicherten Prüfsumme verglichen.

LSIC06.F.3.4.2 Integritätsschutz von Programmen
Die Integrität der Programme wird mit Hilfe kryptografischer Prüfsummenbildung sichergestellt. Die kryptografischen Prüfsummen werden hierbei vor jedem Programmaufruf durch denselben Verschlüsselungsalgorithmus neu berechnet und mit der gespeicherten Prüfsumme verglichen. Ist der Vergleich erfolgreich, wird das Programm ausgeführt.

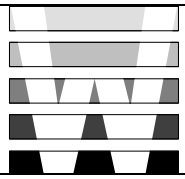
LSIC06.F.3.4.3 Integritätsschutz von Nachrichten
Wie für Daten kann die kryptografische Prüfsummenbildung auch für die Sicherstellung der Integrität von Nachrichten verwendet werden.

LSIC06.F.3.5 Übertragungsnachweis

LSIC06.F.3.5.1 Sendenachweis
Der Empfänger kann nachweisen, daß ihm ein bestimmter Datenstrom von einem eindeutig identifizierbaren Sender übersandt worden ist.

LSIC06.F.3.5.2 Empfangsnachweis
Der Sender eines Datenstromes kann nachweisen, daß dieser Datenstrom seinen Adressaten erreicht hat. Hierzu erhält der Sender eine fälschungssichere Empfangsquittung (z. B. elektronische Unterschrift).

LSIC06.F.4 Entschlüsselung
Es ist möglich, verschlüsselte Daten, Nachrichten oder Programme zu entschlüsseln.



LSIC06.F.5 Information Labeling
Beim Information Labeling werden Objekte und Subjekte, unabhängig von der Zugriffskontrolle, mit einem komplexen Informations-Attribut versehen. Das Attribut eines Subjekts/Objekts geht in das Attribut eines anderen Subjekts/Objekts mit ein, wenn ein Informationsfluß zwischen beiden stattfindet.

Diese Maßnahme sollte eingesetzt werden, wenn die Historie eines Objekts erkennbar sein soll. Der Label wird nicht für die Zugriffskontrolle, sondern vor allem zum Schutz gegen Bedrohungen der Integrität eingesetzt. Der Information Label kann z. B. folgende Informationen enthalten: Herkunft, Typ und Entstehungszeit der Daten, Funktion bzw. Bedeutung der Daten, Vertrauenswürdigkeit der Daten (z. B. aktuelle tatsächliche Einstufung), Liste der Nutzer der Daten (z. B. Nutzer, die einen Schreib/Lese-Zugriff ausgeführt haben), Warnungen bzw. Hinweise zur Handhabung der Daten, Versionsnummern und Hinweise auf Rechte oder Hersteller, Empfehlungen für Einstellungen der nutzerbestimmbaren und regelbasierten Zugriffskontrolle.

LSIC06.F.6 Verdeckte Kanäle
Es sind weder speicher- noch zeitmodulierte Kanäle bekannt, deren Bandbreite ein nichtakzeptables Maß überschreiten, und die Informationen zwischen Prozessen ohne Prüfung der Zugriffsrechte (d. h. verdeckt) übertragen können.

LSIC06.F.7 Schutz vor Inferenz
Es ist nicht möglich, aus verschiedenen niedriger eingestuften Informationen höher eingestufte Informationen abzuleiten.

LSIC06.F.8 Virenerkennung

LSIC06.F.8.1 Virensuche
Es besteht die Möglichkeit des Patternmatchings bezüglich bekannter Viren-Zeichenketten.

LSIC06.F.8.2 Automatische Aktivierung der Virenerkennung
Es kann eingestellt werden, daß die Virensuche oder das Erkennen von Programmänderungen zu bestimmten Zeitpunkten/Ereignissen automatisch aktiviert wird.

LSIC06.F.8.3 Reparatur Viren-befallener Dateien
Wird ein Virus entdeckt, ist es möglich, die befallenen Dateien wieder zu reparieren.

3.5.6.3.4 Sonstige Anforderungen

keine

3.5.7 Leistungseinheit LSIC07 „Zuverlässigkeit der Dienstleistung“

3.5.7.1 Zuordnung zum V-Modell und zur Methodenzuordnung

entfällt

3.5.7.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an die Verfügbarkeit des Systems zusammengefaßt, die sicherstellen, daß Aufgaben genau zu dem erforderlichen Zeitpunkt durchgeführt werden können, daß ein Zugriff in dem erforderlichen Moment möglich ist und daß Betriebsmittel nicht unnötig angefordert oder zurückgehalten werden.

Darin eingeschlossen sind alle Funktionen, die dafür sorgen, daß auf Anforderung einer autorisierten Stelle (z. B. ein Nutzer oder ein Prozeß, der einen Nutzerauftrag ausführt) die Betriebsmittel zugänglich und benutzbar sind und alle Funktionen, die die Beeinflussung von zeitkritischen Operationen verhindern oder einschränken.

Die Leistungseinheit umfaßt alle Funktionen, die

- für die Fehlerentdeckung und -überbrückung,
- zur Begrenzung der Auswirkungen von Fehlern auf den Betrieb,
- zur Minimierung von Unterbrechungen und Leistungsverlusten,
- zur Reaktion auf externe Anforderungen und Ausgaben innerhalb festgelegter Zeiten

vorgesehen sind.

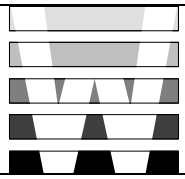
3.5.7.3 Anforderungen

3.5.7.3.1 Anforderungen an die Schnittstellen

keine

3.5.7.3.2 Anforderungen an die Methodenunterstützung

keine



3.5.7.3.3 Anforderungen an die Funktionen

LSIC07.F.1 Fehlerüberbrückung

LSIC07.F.1.1 Fehlertolerante Hard- und Software
Bei Auftreten von Softwarefehlern soll der Fehler überbrückt werden, d. h. die Funktionalität der anderen Funktionen darf dadurch nicht beeinträchtigt werden, sofern dies möglich ist. Widersprüche bei der Steuerung der Leistungseinheiten durch den Nutzer werden von der Software erkannt. Es existieren Fehlerbehandlungsroutinen, und auf die Fehler in den Komponenten der SEU wird kontrolliert reagiert. Der Ausfall bestimmter einzelner Hardwarekomponenten (z. B. Festplatte) wird so überbrückt, daß alle davon unabhängigen Funktionen auch weiterhin kontinuierlich zur Verfügung stehen.

LSIC07.F.1.2 Redundante Auslegung der Hard- und Software
Es existieren redundante Komponenten, die bei Auftreten eines Fehlers in einer Hard- oder Softwarekomponente deren Aufgaben übernehmen. Bei der redundanten Auslegung der Hardware existieren Fehlerkorrekturverfahren, die ein sicheres Weiterarbeiten bei Ausfall von Hardwarekomponenten gestatten.

LSIC07.F.2 Beschränken der Zugriffe von Nutzern auf kritische Ressourcen
Die Zugriffe auf Ressourcen können für einzelne Nutzer/Prozesse durch den Systemverwalter beschränkt werden.

Diese Leistungseinheit soll sowohl vor zielgerichteten Attacken („Denial of service attacks“) als auch vor Fehlern schützen, die sich aus anderen Gründen ergeben (z. B. Adressierungsfehler). Diese Attacken/Fehler können die Zuteilung der Rechenzeit und die Erzeugung von Prozessen, den Verbrauch von Haupt- und Plattenspeicher bzw. von Dateieinträgen sowie die Verkehrsbelastung des Netzwerks betreffen. Um den Verbrauch von Speicherplatz und Rechenzeit zu beschränken, sind z. B. in UNIX für Nutzerprozesse Obergrenzen („Hard limits“) für die Ressourcen-Ausschöpfung festgesetzt: cpu-time, file size, data/stack segment size, core dump size (Speicherauszug), resident set size (Hauptspeicher), number of file descriptors (Anzahl Dateiverweise). Für die Kontrolle des Plattenspeicherplatzes sind „Disk Quotas“ vorgesehen. Hier kann pro Nutzer der verfügbare Platz in Blöcken und die Anzahl der Dateieinträge festgelegt werden.

LSIC07.F.3 Gewährleistung der Verklemmungsfreiheit
Das System muß unabhängig von seiner momentanen Auslastung für bestimmte Aktionen gewährleisten, daß eine Obergrenze für die Reaktionszeit nicht überschritten wird. Dazu gehört auch, daß aufgetretene Verklemmungen rechtzeitig gelöst werden.

Dies kann z. B. durch geeignete Scheduling-Algorithmen erreicht werden.

LSIC07.F.4 Regelmäßige Zwischenspeicherung
Es ist möglich, das System so einzustellen, daß eine regelmäßige Zwischenspeicherung der gerade bearbeiteten Daten automatisch durchgeführt wird.

LSIC07.F.5 Datensicherung

LSIC07.F.5.1 Periodische Datensicherung
Um die Verfügbarkeit der Daten sicherzustellen, wird eine regelmäßige Datensicherung auf externe Datenträger durchgeführt. Es besteht die Möglichkeit der Gesamtsicherung und der Deltasicherung.

LSIC07.F.5.2 Automatik
Es ist möglich, das Verfahren zur Datensicherung ohne manuelle Eingriffe durchzuführen (Zeitpunkte, Umfang und Datenträger werden voreingestellt). Die beschriebenen Datenträger werden auf die Wiedereinlesbarkeit geprüft.

LSIC07.F.5.3 Nutzer-Datensicherung
Es ist möglich, die Datenbasis insgesamt oder in Teilen aufgrund von Nutzeranforderungen zu sichern.

LSIC07.F.6 Verfahren zur Wiedereinspielung gesicherter Daten
Daten, die gesichert worden sind, können wiedereingespielt werden.

LSIC07.F.7 Schutz des Systembetriebs
Um sicherzustellen, daß Nutzer den Server nicht unbefugt außer Betrieb setzen, müssen Zugriffsrechte auf die Systemkommandos zum „Shutdown“ entsprechend gesetzt sein.

Dies alleine ist aber nicht ausreichend, da der Betrieb z. B. auch durch das Abstecken der Konsole oder bei den meisten Systemen durch eine bestimmte Tastenkombination abgebrochen werden kann. Dies muß soweit wie möglich unterbunden sein.

3.5.7.3.4 Sonstige Anforderungen

keine

3.6 V-Modell: Leistungskomplex SE

3.6.1 Leistungseinheit LSE01 „Anforderungen erfassen“

3.6.1.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- SE 1.2 Anwendungssystem beschreiben („Anwenderforderungen.*Grobe Systembeschreibung*“, „Anwenderforderungen.*IT-Sicherheitsziel*“)
- SE 1.3 Kritikalität und Anforderungen an die Qualität definieren („Anwenderforderungen.*Kritikalität des Systems*“, „Anwenderforderungen.*Qualitätsforderungen*“)
- SE 1.4 Randbedingungen definieren („Anwenderforderungen.*Randbedingungen*“)
- SE 1.5 System fachlich strukturieren („Anwenderforderungen.*Fachliche Anforderungen*“)
- SE 1.6 Bedrohung und Risiko analysieren („Anwenderforderungen.*Bedrohungs- und Risikoanalyse*“, „Anwenderforderungen.*IT-Sicherheit*“)
- SE 2.1 System technisch entwerfen („Technische Anforderungen“)
- SE 2.4 Anwenderforderungen zuordnen („Systemarchitektur.*Anforderungszuordnung*“)
- SE 3.1 Allgemeine Anforderungen aus Sicht der SW-/HW-Einheit definieren („Technische Anforderungen.*Allgemeine Anforderungen*“)
- SE 3.2 Anforderungen an die externen Schnittstellen der SW-/HW-Einheit präzisieren („Technische Anforderungen.*Externe Schnittstellen*“)
- SE 3.3 Anforderungen an die Funktionalität definieren („Technische Anforderungen.*Gesamtfunktion*“)
- SE 3.4 Anforderungen an die Qualität der SW-/HW-Einheit definieren („Technische Anforderungen.*Qualitätsforderungen*“)
- SE 3.5 Anforderungen an Entwicklungs- und SWPÄ-Umgebung definieren („Technische Anforderungen.*Anforderungen an Entwicklungs- und SWPÄ-Umgebung*“)
- SE 4.1-SW SW-Architektur entwerfen („SW-Architektur.*Anforderungszuordnung*“)

Methode

keine

3.6.1.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen

- fachliche und technische Anforderungen erfaßt und dokumentiert werden,
- Anforderungen klassifiziert und hierarchisch gruppiert werden,
- Anforderungen den Erzeugnissen, Schnittstellen sowie den Komponenten der Entwicklungs- und Zielrechnerumgebung zugeordnet werden.

Die erfaßten und dokumentierten Anforderungen beziehen sich nicht auf Werkzeuge, die eine oder mehrere Methoden unterstützen. Anforderungen an Werkzeuge, die ein oder mehrere Methoden unterstützen, werden in eigenen Leistungseinheiten aufgeführt.

3.6.1.3 Anforderungen

3.6.1.3.1 Anforderungen an die Schnittstellen

LSE01.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.6.1.3.2 Anforderungen an die Methodenunterstützung

keine

3.6.1.3.3 Anforderungen an die Funktionen

LSE01.F.1 Erfassen von Anforderungen

LSE01.F.1.1 Formulargestützt
Es ist möglich, Anforderungen formulargestützt zu erfassen. Zu jeder Anforderung werden Felder ausgefüllt. Hierbei kann es sich auch um Felder für Freitext und Grafik handeln.

LSE01.F.1.2 Grafisch
Es ist möglich, Anforderungen grafisch zu erfassen.

Beispielsweise können Anforderungen an die Funktionsabläufe sowie datenorientierte Anforderungen mit Hilfe eines speziellen Grafiksystems (mit Symbolen fester Bedeutung) erfaßt werden.

LSE01.F.1.3 Anpaßbarkeit
Die Formulare zur Erfassung von Anforderungen sind projektspezifisch anpaßbar.

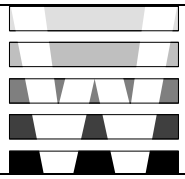
LSE01.F.2 Anforderungszuordnung

LSE01.F.2.1 Segmente
Es ist möglich, eine Anforderung Segmenten zuzuordnen.

LSE01.F.2.2 SW-Einheit
Es ist möglich, eine Anforderung SW-Einheiten zuzuordnen.

LSE01.F.2.3 Schnittstellen
Es ist möglich, eine Anforderung Schnittstellen zuzuordnen.

LSE01.F.2.4 Entwicklungsumgebung
Es ist möglich, eine Anforderung der Entwicklungsumgebung und deren Komponenten zuzuordnen.



LSE01.F.2.5 Zielumgebung
Es ist möglich, eine Anforderung der Zielumgebung zuzuordnen.

LSE01.F.2.6 Anforderungsmengen
Es ist möglich, diejenigen Anforderungen anzeigen zu lassen, die vorgegebenen Segmenten, SW-Einheiten, Schnittstellen, Entwicklungsumgebungen und Zielumgebungen zugeordnet sind.

LSE01.F.3 Bezeichnung von Anforderungen

LSE01.F.3.1 Alphanumerisch
Jede Anforderung bekommt eine alphanumerische Bezeichnung.

LSE01F.3.2 Strukturierung
Mit Hilfe der Bezeichnung ist eine Klassifizierung der Anforderungen sowie deren Einordnung in eine Hierarchie von Anforderungen möglich. Durch diese Einordnung ist eine hierarchische Verfeinerung der Anforderungen darstellbar.

LSE01.F.4 Beschreiben von Anforderungen
Jede Anforderung wird mit folgenden Informationen versehen: Ersteller (Autor), Datum und Kommentar zur Anforderung.

Als Kommentar zu einer Anforderung ist z. B. ein Hinweis möglich, ob die Anforderung ein Vertragsbestandteil oder eine abgeleitete Anforderung ist.

3.6.1.3.4 Sonstige Anforderungen

LSE01.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE01.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE01.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.2 Leistungseinheit LSE02 „Spezifikation von Nutzerschnittstellen unterstützen“

3.6.2.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- SE 2.5 Schnittstellen beschreiben („Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)
- SE 3.2 Anforderungen an die externen Schnittstellen der SW-/HW-Einheit präzisieren („Technische Anforderungen.*Technische Anforderungen an die Nutzerschnittstelle*“)
- SE 4.2-SW SW-interne und -externe Schnittstellen entwerfen („Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)

Methode

- Dialog Design Modellierung (DIAL)

3.6.2.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen

- Inhalte und Bedienungselementen von Nutzerschnittstellen sowie deren dynamisches Verhalten definiert und
- die Strukturen von Nutzerdialogen mit Bildschirmmasken festgelegt werden können.

3.6.2.3 Anforderungen

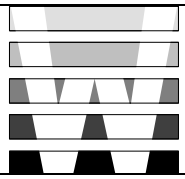
3.6.2.3.1 Anforderungen an die Schnittstellen

LSE02.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE02.S.2 Eingangs-Schnittstelle zu LSE20 „Code rückwärts transformieren“
Es ist möglich, die Spezifikationen für Schnittstellen, die mit LSE20 aus den physischen Schemaspezifikationen erzeugt wurden, ohne weitere Transformation aus der Objektverwaltung zu übernehmen.

LSE02.S.3 Ausgangs-Schnittstelle zu LSE10 „Spezifikation der Komponenten und Module unterstützen“
Es ist möglich, die Spezifikationen für Schnittstellen ohne weitere Transformation über die Objektverwaltung an LSE10 zur Einbindung in die Spezifikation von Komponenten und Modulen zu übergeben.

LSE02.S.4 Ausgangs-Schnittstelle zu LSE14 „Nutzerschnittstellenformate generieren“
Es ist möglich, die Spezifikationen für Schnittstellen ohne weitere Transformation an LSE14 zur Generierung von physischen Schemaspezifikationen für Schnittstellen zu übergeben.



LSE02.S.5 Ausgangs-Schnittstelle zu LSE19 „Zeitverhalten simulieren“
Es ist möglich, die Spezifikationen für Schnittstellen ohne weitere Transformation über die Objektverwaltung an LSE19 zur Simulation des voraussichtlichen Zeitverhaltens zu übergeben.

LSE02.S.6 Ausgangs-Schnittstelle zu LQS14 „Prüffälle generieren“
Es ist möglich, die Spezifikationen für Schnittstellen ohne weitere Transformation über die Objektverwaltung an LQS14 zur Generierung von Prüffällen zu übergeben.

3.6.2.3.2 Anforderungen an die Methodenunterstützung

LSE02.M.1 Dialog Design Modellierung (DIAL)

LSE02.M.1.1 Symbole

LSE02.M.1.1.1 Maskentypen
Es stehen Symbole zur Darstellung aller für einen Dialog notwendigen Maskentypen zur Verfügung.

LSE02.M.1.1.2 Zustandssymbole
Es stehen Symbole zur Darstellung der mit den einzelnen Maskentypen verbundenen Zustände zur Verfügung. Es ist möglich, diese Symbole textuell zu kennzeichnen.

Hierbei kann es sich z. B. um textuell gekennzeichnete horizontale Linien handeln, denen Maskentypen vorangestellt sind.

LSE02.M.1.1.3 Übergangssymbole
Es stehen Symbole zur Darstellung des Übergangs von einem Zustand zum anderen zur Verfügung.

LSE02.M.1.2 Zustandsübergänge

LSE02.M.1.2.1 Alternative Zustandsübergänge
Es ist möglich, in Dialog-Diagrammen alternative Zustandsübergänge darzustellen.

LSE02.M.1.2.2 Bedingte Zustandsübergänge
Es ist möglich, in Dialog-Diagrammen bedingte Zustandsübergänge darzustellen.

3.6.2.3.3 Anforderungen an die Funktionen

LSE02.F.1 Behandlung von konventionellen Nutzerschnittstellen

LSE02.F.1.1 Angabe Zielsystem

LSE02.F.1.1.1 Masken
Es ist möglich, für Masken das Zielsystem anzugeben.

LSE02.F.1.1.2 Listen
Es ist möglich, für Listen das Zielsystem anzugeben.

LSE02.F.1.2 Angabe der Größe

LSE02.F.1.2.1 Masken
Es ist möglich, für Masken die Größe in Zeilen und Spalten anzugeben.

LSE02.F.1.2.2 Liste
Es ist möglich, für Listen die Größe in Zeilen und Spalten anzugeben.

LSE02.F.1.3 Grafische Elemente

LSE02.F.1.3.1 Darstellung
Es ist möglich, grafische Elemente zumindest als waagerechte und senkrechte Striche darzustellen.

LSE02.F.1.3.2 Feldattribute
Es ist möglich, grafische Elemente mit Feldattributen zu versehen.

LSE02.F.1.4 Literale

LSE02.F.1.4.1 Position
Es ist möglich, Literale auf beliebiger Position darzustellen.

LSE02.F.1.4.2 Feldattribute
Es ist möglich, Literale mit Feldattributen zu versehen.

LSE02.F.1.5 Variable

LSE02.F.1.5.1 Position
Es ist möglich, eine Variable auf beliebiger Position darzustellen.

Variable sind Ein- oder Ausgabefelder.

LSE02.F.1.5.2 Feldattribute
Es ist möglich, eine Variable mit Feldattributen zu versehen.

Als Feldattribute sind u. a. die Attribute „geschützt“, „unterstrichen“, usw. zu versehen.

LSE02.F.1.6 Attributzuweisung

LSE02.F.1.6.1 Gezielte Auswahl
Es ist möglich, bei der Zuweisung von Feldattributen eine zielsystemgerechte Auswahl zu erhalten.

LSE02.F.1.6.2 Voreinstellung
Es ist möglich, eine Voreinstellung der Feldattribute vorzunehmen.

LSE02.F.1.6.3 Änderung der Voreinstellung
Es ist möglich, eine gültige Voreinstellung zu ändern, ohne die Voreinstellungen in ihrer globalen Wirksamkeit aufzuheben.

LSE02.F.1.7 Interaktionssteuerung

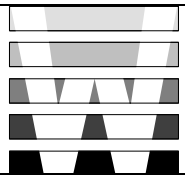
LSE02.F.1.7.1 Cursor
Es ist möglich, das erste Eingabefeld zu bestimmen.

LSE02.F.1.7.2 Funktionstasten
Es ist möglich, Funktionstasten mit Werten zu versehen.

LSE02.F.1.8 Datendefinition

LSE02.F.1.8.1 Global
Es ist möglich, bei der Einstellung einer Variablen auf global gültige Datenstrukturen zuzugreifen.

LSE02.F.1.8.2 Lokal
Es ist möglich, bei der Einstellung einer Variablen deren Gültigkeit auf das Objekt (Maske oder Liste) zu beschränken.



LSE02.F.1.9 Vorschau

LSE02.F.1.9.1 Masken
Es ist möglich, eine Maske in der Umgebung des Ziel-TP-Monitors darzustellen.

LSE02.F.1.9.2 Listen
Es ist möglich, eine Darstellung zu wählen, die der Umgebung des Zielsystems entspricht.

LSE02.F.1.10 Kopieren

LSE02.F.1.10.1 Masken
Es ist möglich, eine Maske mit allen anhängenden Informationen zu kopieren.

LSE02.F.1.10.2 Listen
Es ist möglich, eine Liste mit allen anhängenden Informationen zu kopieren.

LSE02.F.1.11 Modulzuweisung
Es ist möglich, Objekten Verarbeitungsmodule zuzuordnen.
Objekte sind Masken, Listen, Variable, usw.

LSE02.F.1.12 Listenaufbau

LSE02.F.1.12.1 Kopfzeile
Es ist möglich, für eine Liste in einem gesonderten Bereich Kopfzeileninformationen zu definieren.

LSE02.F.1.12.2 Fußzeile
Es ist möglich, für eine Liste in einem gesonderten Bereich Fußzeileninformationen zu definieren.

LSE02.F.1.13 Gruppenwechsel

LSE02.F.1.13.1 Einfacher Gruppenwechsel
Es ist möglich, für eine Liste einen einstufigen Gruppenwechsel zu definieren.

LSE02.F.1.13.2 Geschachtelter Gruppenwechsel
Es ist möglich, für eine Liste geschachtelte Gruppenwechsel zu definieren.

LSE02.F.1.13.3 Auslöser
Es ist möglich, als Auslöser für einen Gruppenwechsel mehrere variable Feldinhalte zusammenzufassen.

LSE02.F.2 Behandlung von fensterorientierten Nutzerschnittstellen

LSE02.F.2.1 Fenster

LSE02.F.2.1.1 Fensterarten

LSE02.F.2.1.2 Einzel-Fenster-Schnittstelle
Es ist möglich, einem Fenster den Typ „Einzel-Fenster-Schnittstelle“ zuzuordnen.
Ein Fenster vom Typ „Einzel-Fenster-Schnittstelle“ steht allein und kontrolliert innerhalb seines Fensterrahmens keine weiteren abhängigen Fenster.

LSE02.F.2.1.3 Vielfach-Fenster-Schnittstelle
Es ist möglich, einem Fenster den Typ „Vielfach-Fenster-Schnittstelle“ zuzuordnen.
Ein Fenster vom Typ „Vielfach-Fenster-Schnittstelle“ kontrolliert als Primärfenster innerhalb seines Fensterrahmens weitere abhängige Fenster.

LSE02.F.2.1.4 Modale Fenster
Es ist möglich, ein Fenster als modal zu definieren.

Ein modales Fenster erwartet unbedingt eine Reaktion des Nutzers, bevor der Dialog weitergeführt wird. Eine parallele Weiterarbeit des Nutzers an einem anderen Fenster wird hierdurch ausgeschlossen.

LSE02.F.2.2 Fensterpräsentation

LSE02.F.2.2.1 Skalierbarer Fensterrahmen
Es ist möglich, ein Fenster mit einem größenveränderbaren (skalierbarem) Rahmen zu versehen.

LSE02.F.2.2.2 Starrer Fensterrahmen
Es ist möglich, ein Fenster mit einem Rahmen von starrer Größe zu versehen.

Pop-up-Fenster besitzen in der Regel einen starren Fensterrahmen.

LSE02.F.2.2.3 Fenstertitel
Es ist möglich, ein Fenster mit einem Titel zu versehen.

LSE02.F.2.2.4 Vertikaler Verschiebebalken
Es ist möglich, ein Fenster mit einem vertikalen Verschiebebalken zu versehen.

LSE02.F.2.2.5 Horizontaler Verschiebebalken
Es ist möglich, ein Fenster mit einem horizontalen Verschiebebalken zu versehen.

LSE02.F.2.2.6 Schließbox
Es ist möglich, ein Fenster mit einer Schließbox zu versehen.

Bei Betätigung der Schließbox wird das Fenster geschlossen und vom Bildschirm entfernt.

LSE02.F.2.2.7 Minimierungsbox
Es ist möglich, ein Fenster mit einer Minimierungsbox zu versehen.

Bei Betätigung der Minimierungsbox wird das Fenster ikonisiert.

LSE02.F.2.2.8 Fensterikone
Es ist möglich, einem Fenster für die Minimierung eine Ikone zuzuordnen.

Die zugeordnete Ikone repräsentiert das Fenster, nachdem die Minimierungsbox betätigt wurde.

LSE02.F.2.2.9 Maximierungsbox
Es ist möglich, ein Fenster mit einer Maximierungsbox zu versehen.

Nach Betätigung der Maximierungsbox füllt das Fenster den gesamten Bildschirm aus.

LSE02.F.2.2.10 Befehlsleiste
Es ist möglich, ein Fenster mit einer Befehlsleiste zu versehen.

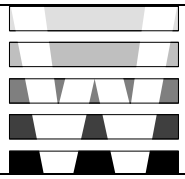
LSE02.F.2.2.11 Vordefinierte Ereignisse für Fenster
Es ist möglich, einem Fenster vordefinierte Ereignisse zuzuordnen.

Vordefinierte Ereignisse sind u. a. Mausbewegungen, Betätigen von Maustasten, Größenveränderungen, Aktivieren und Deaktivieren von Fenstern.

LSE02.F.2.3 Befehlsleisten

LSE02.F.2.3.1 Vordefinierte Pull-Down-Menüs
Es ist möglich, vordefinierte Pull-Down-Menüs in Befehlsleisten einzubinden.

Vordefinierte Pull-Down-Menüs enthalten z. B. grundlegende fensterorientierte Befehle wie Wiederherstellen, Verschieben, Größe verändern, Schließen des aktiven



Fensters, usw. oder enthalten Befehle für Hilfefunktionen wie Hilfe zur Hilfe, Hilfeindex, Hilfen zu Funktionstasten.

LSE02.F.2.3.2 Befehle definieren
Es ist möglich, Befehle zu definieren.

LSE02.F.2.3.3 Codeabschnitt zu Befehlen
Es ist möglich, Befehle mit einem Codeabschnitt zu hinterlegen.

LSE02.F.2.3.4 Befehle hierarchisch gruppieren
Es ist möglich, Befehle hierarchisch zu gruppieren, wobei jede Verfeinerungsstufe eines Hierarchieastes einem Pull-Down-Menü innerhalb der Befehlsleiste entspricht.

LSE02.F.2.3.5 Befehle ändern
Es ist möglich, Befehle zu ändern.

LSE02.F.2.3.6 Befehle verschieben
Es ist möglich, Befehle zu verschieben.

LSE02.F.2.3.7 Befehle löschen
Es ist möglich, Befehle zu löschen.

LSE02.F.2.3.8 Befehlsmodus deaktiviert
Es ist möglich, Befehlen den Modus „deaktiviert“ zuzuordnen.

In diesem Fall sind die entsprechenden Befehle nicht ausführbar. Die Deaktivierung wird am Bildschirm durch eine optisch veränderte Darstellung kenntlich gemacht. In der Regel wird der Befehlstext anstelle in normalem Farbton in einem schemenhaften Grauton angezeigt.

LSE02.F.2.3.9 Funktionstasten/Tastenkombinationen
Es ist möglich, Befehle mit Funktionstasten bzw. einer Tastenkombination zu hinterlegen.

Der entsprechende Befehl wird dann durch Betätigen der hinterlegten Funktionstaste bzw. Tastenkombination aktiviert.

LSE02.F.2.3.10 Mnemonische Befehlskürzel
Es ist möglich, Befehle mit einem mnemonischen Befehlskürzel zu hinterlegen.

Der entsprechende Befehl wird dann durch Eingabe des mnemonischen Befehlskürzels aktiviert. In der Regel ist dies ein durch Unterstreichen besonders gekennzeichneter Buchstabe aus dem Befehl selbst.

LSE02.F.2.3.11 Befehle in Bibliothek kopieren
Es ist möglich, Befehle in eine Bibliothek zu kopieren.

LSE02.F.2.3.12 Befehle aus Bibliothek kopieren
Es ist möglich, Befehle aus einer Bibliothek in eine Befehlsleiste zu kopieren.

LSE02.F.2.4 Kontrollelement

LSE02.F.2.4.1 Statische Kontrollelemente

LSE02.F.2.4.1.1 Texte
Es ist möglich, statische Texte als Kontrollelemente auf einer Fensterfläche einzufügen.

LSE02.F.2.4.1.2 Gruppenrahmen
Es ist möglich, Gruppenrahmen als Kontrollelemente auf einer Fensterfläche einzufügen.

LSE02.F.2.4.1.3 Ikonen
Es ist möglich, Ikonen als Kontrollelemente auf einer Fensterfläche einzufügen.

LSE02.F.2.4.1.4 Bitmaps
Es ist möglich, Bitmaps als Kontrollelemente auf einer Fensterfläche einzufügen.

LSE02.F.2.4.2 Schalter

LSE02.F.2.4.2.1 Push-Buttons
Es ist möglich, Push-Buttons als Kontrollelemente auf einer Fensterfläche einzufügen.

LSE02.F.2.4.2.2 Radio-Buttons
Es ist möglich, Radio-Buttons als Kontrollelemente auf einer Fensterfläche einzufügen.

LSE02.F.2.4.2.3 Check-Boxes
Es ist möglich, Check-Boxes als Kontrollelemente auf einer Fensterfläche einzufügen.

LSE02.F.2.4.2.4 Vertikale Verschiebebalken
Es ist möglich, vertikale Verschiebebalken als Kontrollelemente auf einer Fensterfläche einzufügen.

LSE02.F.2.4.2.5 Horizontale Verschiebebalken
Es ist möglich, horizontale Verschiebebalken als Kontrollelemente auf einer Fensterfläche einzufügen.

LSE02.F.2.4.2.6 Hot Spots
Es ist möglich, Hot Spots als Kontrollelemente auf einer Fensterfläche einzufügen.

LSE02.F.2.4.3 Felder

LSE02.F.2.4.3.1 Eingabefelder
Es ist möglich, Eingabefelder als Kontrollelemente auf einer Fensterfläche einzufügen.

LSE02.F.2.4.3.2 Eingabefeldlisten
Es ist möglich, Eingabefeldlisten als Kontrollelemente auf einer Fensterfläche einzufügen.

Eingabefeldlisten dienen zur Erfassung von mehrzeiligen Eingaben. Mit Verschiebebalken kann innerhalb der Eingabefeldliste navigiert werden.

LSE02.F.2.4.3.3 List-Boxes
Es ist möglich, List-Boxes als Kontrollelemente auf einer Fensterfläche einzufügen.

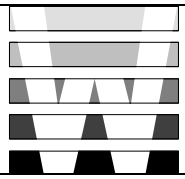
List-Boxes dienen zum Anzeigen von mehrzeiligen Ausgaben. Mit Verschiebebalken kann innerhalb der Anzeige navigiert werden.

LSE02.F.2.4.3.4 Combo-Boxes
Es ist möglich, Combo-Boxes als Kontrollelemente auf einer Fensterfläche einzufügen.

Eine Combo-Box kombiniert die Funktionen eines Eingabefelds mit einer List-Box, in der die jeweiligen Auswahlmöglichkeiten entweder permanent oder erst explizit nach Betätigung eines Drop-Down-Schalters angezeigt werden.

LSE02.F.2.4.3.5 Datentypen
Es ist möglich, Feldern Datentypen zuzuordnen.

LSE02.F.2.4.3.6 Kontrollelemente selektieren
Es ist möglich, Kontrollelemente zu selektieren.



LSE02.F.2.4.3.7 Kontrollelemente verschieben
Es ist möglich, selektierte Kontrollelemente zu verschieben.

LSE02.F.2.4.3.8 Kontrollelemente löschen
Es ist möglich, selektierte Kontrollelemente zu löschen.

LSE02.F.2.4.3.9 Kontrollelemente kopieren
Es ist möglich, selektierte Kontrollelemente zu kopieren.

LSE02.F.2.4.3.10 Präsentationsparameter festlegen
Es ist möglich, für selektierte Kontrollelemente Präsentationsparameter festzulegen.

Präsentationsparameter sind u. a. Schriftart/Schriftgröße, Vordergrundfarbe, Hintergrundfarbe.

LSE02.F.2.4.3.11 Kontrollelemente zusammenfassen
Es ist möglich, selektierte Kontrollelemente zu Gruppen zusammenzufassen.

LSE02.F.2.4.3.12 Kontrollelemente bündig ausrichten
Es ist möglich, selektierte Kontrollelemente an einem Raster linksbündig und rechtsbündig auszurichten.

LSE02.F.2.4.3.13 Kontrollelemente horizontal distanzieren
Es ist möglich, selektierte Kontrollelemente in gleicher horizontaler Distanz zueinander auszurichten.

LSE02.F.2.4.3.14 Kontrollelemente vertikal distanzieren
Es ist möglich, selektierte Kontrollelemente in gleicher vertikaler Distanz zueinander auszurichten.

LSE02.F.2.4.3.15 Distanz zwischen Kontrollelementen festlegen
Es ist möglich, die gewünschte vertikale und horizontale Distanz zwischen Kontrollelementen festzulegen.

LSE02.F.2.4.3.16 Größe von Kontrollelementen angleichen
Es ist möglich, die Größe selektierter Kontrollelemente automatisch anzugleichen.

LSE02.F.2.4.3.17 Präsentationsparameter von Kontrollelementen angleichen
Es ist möglich, die Präsentationsparameter selektierter Kontrollelemente automatisch anzugleichen.

LSE02.F.2.4.3.18 Kontrollelemente in Bibliothek kopieren
Es ist möglich, auf einer Fensterfläche selektierte Kontrollelemente in eine Bibliothek zu kopieren.

LSE02.F.2.4.3.19 Kontrollelemente aus Bibliothek kopieren
Es ist möglich, aus einer Bibliothek Kontrollelemente auf eine Fensterfläche zu kopieren.

LSE02.F.2.4.4 Ereignisse

LSE02.F.2.4.4.1 Vordefinierte Ereignisse für Kontrollelemente
Es ist möglich, Kontrollelementen vordefinierte Ereignisse zuzuordnen.

Vordefinierte Ereignisse sind u. a. Betätigen von Maustasten, Auswahl von Listeinträgen, Beendigen von Feldeingaben, Aktivieren und Deaktivieren von Schaltern.

LSE02.F.2.4.4.2 Codeabschnitt zu Ereignis
Es ist möglich, jedes Ereignis mit einem Codeabschnitt zu hinterlegen.

3.6.2.3.4 Sonstige Anforderungen

LSE02.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE02.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE02.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.3 Leistungseinheit LSE03 „Architekturentwurf unterstützen“

3.6.3.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 4.1-SW SW-Architektur entwerfen
(„SW-Architektur.*Lösungsvorschläge*“,
„SW-Architektur.*Modularisierung/Datenbankentwurf*“)

Methode

– Objektentwurfstechnik (OBJE)

– Structured Design (STRD)

3.6.3.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur statischen Zerlegung einer SW-Einheit in SW-Komponenten und SW-Module gestellt. Ebenso werden Anforderungen an Werkzeuge bezüglich der Kommunikation dieser SW-Komponenten und SW-Module in Form von Daten- und Kontrollkopplungen gestellt.

3.6.3.3 Anforderungen

3.6.3.3.1 Anforderungen an die Schnittstellen

LSE03.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE03.S.2 Ausgangs-Schnittstelle zu LSE10 „Spezifikation der Komponenten und Module unterstützen“
Es ist möglich, die in einer Architektur festgelegten Komponenten und Module über die Objektverwaltung an LSE10 als Bezugspunkte für die Spezifikation zu übergeben.

LSE03.S.3 Ausgangs-Schnittstelle zu LSE11 „Spezifikation der Datenbanken unterstützen“
Es ist möglich, die in einer Architektur festgelegten Datenbanken über die Objektverwaltung an LSE11 als Bezugspunkte für die Spezifikation zu übergeben.

3.6.3.3.2 Anforderungen an die Methodenunterstützung

LSE03.M.1 Objektentwurfstechnik (OBJE)

LSE03.M.1.1 Symbole

LSE03.M.1.1.1 Objekt
Es stehen Symbole zur Darstellung von Objekten zur Verfügung.

Aus diesen Symbolen ist auch erkenntlich, ob es sich um ein passives oder aktives Objekt handelt. Aktive Objekte enthalten einen eigenen Kontrollfluß, während passive Objekte nur aktiv werden, wenn eine Operation von anderen Objekten angestoßen wird.

LSE03.M.1.1.2 Hierarchisierung

LSE03.M.1.1.2.1 Besteht aus
Es stehen Symbole zur Verfügung, die die Zerlegung eines Objekts in Teilobjekte darstellen.

LSE03.M.1.1.2.2 Benutzt
Es stehen Symbole zur Verfügung, mit denen dargestellt werden kann, daß ein Objekt Operationen eines anderen Objekts benutzt.

LSE03.M.1.1.3 Datenfluß
Es stehen Symbole zur Darstellung des Datenflusses zwischen Objekten zur Verfügung. Dies schließt auch die Kennzeichnung der Richtung des Datenflusses mit ein.

Solche Datenflüsse können sowohl uni- (nur in eine Richtung laufend) als auch bidirektional (in beide Richtungen laufend) sein.

LSE03.M.1.1.4 Ausnahme
Es stehen Symbole zur Verfügung, mit denen dargestellt werden kann, daß ein benutztes Objekt eine Ausnahme an das benutzende Objekt liefern kann.

Eine Ausnahme stellt einen nicht vorzeitigen Rücksprung während der Ausführung einer Operation des benutzten Objekts dar. Es wird dann automatisch der Ausnahmebehandler des benutzenden Objekts abgearbeitet. Dieses Konzept wurde der Programmiersprache Ada entlehnt.

LSE03.M.1.2 Ausblenden
Es ist möglich, Teile des Objektdiagramms nach gewissen Kriterien auszublenden.

Beispielsweise kann nur die oberste Hierarchieebene angezeigt werden oder nur direkt betroffene Objekte eines vorgegebenen Objekts.

LSE03.M.1.3 Namen
Es ist möglich, für Objekte, Operationen und Datenflüsse Namen zu vergeben.

LSE03.M.2 Structured Design (STRD)

LSE03.M.2.1 Symbole

LSE03.M.2.1.1 SW-Komponente/SW-Modul
Es stehen Symbole zur Darstellung der SW-Komponenten und SW-Module zur Verfügung.

LSE03.M.2.1.2 Statische Zerlegung
Es stehen Symbole zur Verfügung, die die Benutzung der SW-Komponenten und SW-Module darstellt.

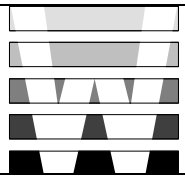
LSE03.M.2.1.3 Kommunikation

LSE03.M.2.1.3.1 Datenkopplung
Es stehen Symbole zur Darstellung der Datenkopplung zur Verfügung.

LSE03.M.2.1.3.2 Kontrollkopplung
Es stehen Symbole zur Darstellung der Kontrollkopplung zur Verfügung.

LSE03.M.2.1.4 Konnektoren
Es stehen Symbole zur Verfügung, um Teilbäume eines Structure Charts auf zusätzliche Seiten der Dokumentation auszulagern.

LSE03.M.2.1.5 Zeitfolgensymbolik
Es stehen Symbole zur Darstellung der zeitlichen Reihenfolge des Aufrufes von Komponenten und Modulen zur Verfügung.



LSE03.M.2.2 Beschreibungsmittel

LSE03.M.2.2.1 Bezeichnung
Die Grafikobjekte werden über Identifikatoren bezeichnet.

LSE03.M.2.2.2 Datenkopplung
Es ist möglich, die Datenkopplung mit Datenstrukturen zu beschreiben.

LSE03.M.2.2.3 Kontrollkopplung
Es ist möglich, die Kontrollkopplung mit Kontrollinformationen zu beschreiben.

3.6.3.3.3 Anforderungen an die Funktionen

LSE03.F.1 Ableitung der Architektur
Es ist möglich, aus der Struktur des Funktionenmodells einen Vorschlag für die Architektur zu erzeugen.

3.6.3.3.4 Sonstige Anforderungen

LSE03.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE03.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.4 Leistungseinheit LSE04 „Prozeßmodellierung unterstützen“

3.6.4.1 Zuordnung zum V-Modell

V-Modell-Aktivität (-Produkt)

– SE 3.3 Anforderungen an die Funktionalität definieren
(„Technische Anforderungen.*Gesamtfunktion*“)

– SE 4.1-SW SW-Architektur entwerfen
(„SW-Architektur.*Lösungsvorschläge*“,
„SW-Architektur.*Modularisierung/Datenbankentwurf*“)

Methode

– Prozeßinteraktionsmodellierung (PZIM)

– Zustandsübergangsmodellierung (ZUST)

3.6.4.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen der Grobentwurf einer SW-Einheit unter Echtzeitgesichtspunkten durchgeführt werden kann. Mit Hilfe eines Netzmodells werden die Strukturierung in sequentielle und parallele Prozesse sowie die Prozeßsynchronisation und -kommunikation beschrieben. Für jeden einzelnen Prozeß werden seine Zustandsübergänge beschrieben. Für miteinander kommunizierende Prozesse werden die Signalflüsse beschrieben.

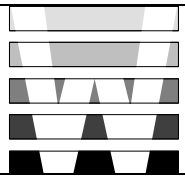
3.6.4.3 Anforderungen

3.6.4.3.1 Anforderungen an die Schnittstellen

LSE04.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE04.S.2 Ausgangs-Schnittstelle zu LSE19 „Zeitverhalten simulieren“
Es ist möglich, Prozeßmodelle ohne weitere Transformation über die Objektverwaltung an LSE19 zur Simulation des voraussichtlichen Zeitverhaltens zu übergeben.

LSE04.S.3 Ausgangs-Schnittstelle zu LQS14 „Prüffälle generieren“
Es ist möglich, Prozeßmodelle ohne weitere Transformation über die Objektverwaltung an LQS14 zur Generierung von Prüffällen zu übergeben.



3.6.4.3.2 Anforderungen an die Methodenunterstützung

LSE04.M.1 Prozeßinteraktionsmodellierung (PZIM)

LSE04.M.1.1 Symbole

LSE04.M.1.1.1 Prozesse
Es stehen Symbole zur Darstellung von Prozessen zur Verfügung.

LSE04.M.1.1.2 Signalflüsse
Es stehen Symbole zur Darstellung von Signalflüssen zur Verfügung.

LSE04.M.1.1.3 Zeitfolgensymbolik
Die zeitliche Reihenfolge der Signalflüsse wird aus der Darstellungsform ersichtlich.

LSE04.M.2 Zustandsübergangsmodellierung (ZUST)

LSE04.M.2.1 Symbole

LSE04.M.2.1.1 Zustände
Es stehen Symbole zur Darstellung von Zuständen zur Verfügung.

LSE04.M.2.1.2 Zustandsübergänge
Es stehen Symbole zur Darstellung von Zustandsübergängen zur Verfügung.

LSE04.M.2.1.3 E/A-Symbolik
Es ist möglich, Ein- (Ereignisse) und Ausgaben (Aktionen) zu vermerken.

LSE04.M.2.2 Teilautomaten
Die Zerlegung in mehrere verbundene oder geschachtelte Teilautomaten ist möglich.

LSE04.M.2.3 Darstellungsmöglichkeiten von Zustandsübergangsdiagrammen

LSE04.M.2.3.1 Bubble-Diagramm
Die Darstellung in Form eines Bubble-Diagramms ist möglich.

LSE04.M.2.3.2 Fence-Diagramm
Die Darstellung in Form eines Fence-Diagramms ist möglich.

LSE04.M.2.3.3 Zustandsübergangstabelle
Die Darstellung in Form einer Zustandsübergangstabelle ist möglich.

LSE04.M.2.3.4 Zustandsübergangsmatrix
Die Darstellung in Form einer Zustandsübergangsmatrix ist möglich.

LSE04.M.2.3.5 Darstellungstransformation
Eine automatische Transformation der Zustandsübergangsdiagramme von einer Darstellungsform in eine gewünschte andere Darstellungsform ist möglich.

3.6.4.3.3 Anforderungen an die Funktionen

LSE04.F.1 Bearbeitung des Prozeßmodells
Es ist möglich, den Standard-Symbolsatz durch eine Definition eigener Symbole zu erweitern oder abzuändern.

LSE04.F.2 Konvertierung
Es ist möglich, das Prozeßmodell von der grafischen in eine alphanumerische Notation zu konvertieren und umgekehrt.

3.6.4.3.4 Sonstige Anforderungen

LSE04.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE04.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE04.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.5 Leistungseinheit LSE05 „Funktionsstrukturierung unterstützen“

3.6.5.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- SE 1.1 Ist-Aufnahme/-Analyse durchführen („Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“)
- SE 1.5 System fachlich strukturieren („Anwenderforderungen.*Grobe Systembeschreibung*“, „Anwenderforderungen.*Beschreibung der Funktionalität*“)
- SE 2.1 System technisch entwerfen („Technische Anforderungen.*Gesamtfunktion*“)
- SE 3.3 Anforderungen an die Funktionalität definieren („Technische Anforderungen.*Gesamtfunktion*“)

Methode

- Funktionale Dekomposition (FKTD)

3.6.5.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen

- Funktionen auf unterschiedlichen Detaillierungsstufen beschrieben,
- die definierten Funktionen hierarchisch zueinander in Beziehung gesetzt werden.

Der hierarchische Zusammenhang zwischen Funktionen wird im Sinne einer „Besteht aus“-Beziehung dargestellt.

3.6.5.3 Anforderungen

3.6.5.3.1 Anforderungen an die Schnittstellen

LSE05.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE05.S.2 Ausgangs-Schnittstelle zu LSE06 „Spezifikation der Funktionen unterstützen“
Es ist möglich, erfaßte Funktionen über die Objektverwaltung an LSE06 als Bezugspunkte für die Spezifikation zu übergeben.

LSE05.S.3 Ausgangs-Schnittstelle zu LSE07 „Modellierung der Informationsflüsse unterstützen“
Erfaßte Funktionen werden ohne weitere Transformation über die Objektverwaltung an LSE07 zur automatischen Einbindung in Datenflußdiagramme übergeben.
Die in LSE05 definierten Funktionen entsprechen den in Datenflußdiagrammen verwendeten Prozessen.

LSE05.S.4 Ausgangs-Schnittstelle zu LSE08 „Modellierung der Funktionsdynamik unterstützen“
Erfaßte Funktionen werden ohne weitere Transformation über die Objektverwaltung an LSE08 zur automatischen Einbindung in Ablaufmodelle übergeben.

Die in LSE05 definierten Funktionen entsprechen den in Ablaufmodellen verwendeten Funktionen.

LSE05.S.5 Ausgangs-Schnittstelle zu LSE09 „Informationsstrukturierung unterstützen“
Es ist möglich, erfaßte Funktionen über die Objektverwaltung an LSE09 als Bezugspunkte für die Erstellung von Informationsstrukturen zu übergeben.

3.6.5.3.2 Anforderungen an die Methodenunterstützung

LSE05.M.1 Funktionale Dekomposition (FKTD)

LSE05.M.1.1 Symbole
Es stehen Symbole zur Darstellung von Funktionen zur Verfügung.

LSE05.M.1.2 Baumstruktur
In der Darstellung der Diagrammelemente wird eine Baumstruktur erzwungen.

3.6.5.3.3 Anforderungen an die Funktionen

LSE05.F.1 Anzeigemöglichkeiten

LSE05.F.1.1 Anzeigetiefe
Es ist möglich, die Anzeigetiefe eines Funktionsbaumes gemäß Vorgabe einzuschränken.

LSE05.F.1.2 Form des Funktionsbaumes
Es ist möglich, die Form des Funktionsbaumes in eine eingerückte oder liegende Darstellung zu überführen.

3.6.5.3.4 Sonstige Anforderungen

LSE05.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE05.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE05.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.6 Leistungseinheit LSE06 „Spezifikation der Funktionen unterstützen“

3.6.6.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- SE 1.1 Ist-Aufnahme/-Analyse durchführen(„Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“)
- SE 1.5 System fachlich strukturieren („Anwenderforderungen.*Beschreibung der Funktionalität*“)
- SE 3.3 Anforderungen an die Funktionalität definieren („Technische Anforderungen.*Gesamtfunktion*“)

Methode

keine

3.6.6.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Spezifikation

- der funktionsinternen Ablauflogik und
- der Datenbehandlung innerhalb einer Funktion

gestellt.

3.6.6.3 Anforderungen

3.6.6.3.1 Anforderungen an die Schnittstellen

LSE06.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE06.S.2 Eingangs-Schnittstelle zu LSE05 „Funktionsstrukturierung unterstützen“
Es ist möglich, mit LSE05 erfaßte Funktionen gezielt aus der Objektverwaltung als Bezugspunkte für die Spezifikation zu übernehmen.

LSE06.S.3 Ausgangs-Schnittstelle zu LSE10 „Spezifikation der Komponenten und Module unterstützen“
Es ist möglich, Spezifikationen für Funktionen ohne weitere Transformation über die Objektverwaltung an LSE10 zur Spezifikation von Komponenten und Modulen zu übergeben.

LSE06.S.4 Ausgangs-Schnittstelle zu LQS14 „Prüffälle generieren“
Es ist möglich, Spezifikationen für Funktionen ohne weitere Transformation über die Objektverwaltung an LQS14 zur Generierung von Prüffällen zu übergeben.

3.6.6.3.2 Anforderungen an die Methodenunterstützung

keine

3.6.6.3.3 Anforderungen an die Funktionen

LSE06.F.1 Darstellungsmittel

LSE06.F.1.1 Struktogramm
Es ist möglich, Funktionen mit Struktogrammen (Aktionsdiagrammen) zu beschreiben.

LSE06.F.1.2 Titelblock
Jede Beschreibung wird durch einen Titelblock eingeleitet.

LSE06.F.1.3 Beschreibung von Kontrollstrukturen
Es ist möglich, als Kontrollstrukturen Sequenz, Selektion und Iteration darzustellen.

LSE06.F.1.4 Datenspeicher

LSE06.F.1.4.1 Darstellung
Es ist möglich, Datenspeicherzugriffe grafisch darzustellen.

LSE06.F.1.4.2 Verknüpfung
Es ist möglich, zu einem Datenspeicher die Verknüpfung zu den entsprechenden Entitäten darzustellen.

LSE06.F.1.5 Kontrollstrukturen
Es stehen Symbole zur Darstellung von Kontrollstrukturen zur Verfügung.

LSE06.F.1.6 Datenzugriffe
Es stehen Symbole zur Darstellung von Datenzugriffen zur Verfügung.

LSE06.F.1.7 Blockstrukturen
Es ist möglich, logisch zusammenhängende Strukturen grafisch darzustellen.

LSE06.F.1.8 Darstellung von Konstrukten
Es ist möglich, Beginn und Ende von Konstrukten grafisch darzustellen.

LSE06.F.1.9 Ausblenden von Strukturen
Es ist möglich, in Struktogrammen (Aktionsdiagrammen) logisch zusammenhängende Strukturen zu extrahieren bzw. zu expandieren.

3.6.6.3.4 Sonstige Anforderungen

LSE06.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE06.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE06.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.7 Leistungseinheit LSE07 „Modellierung der Informationsflüsse unterstützen“

3.6.7.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- SE 1.1 Ist-Aufnahme/-Analyse durchführen
  („Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“)
- SE 1.2 Anwendungssystem beschreiben
  („Anwenderforderungen.*Grobe Systembeschreibung*“)
- SE 1.5 System fachlich strukturieren
  („Anwenderforderungen.*Beschreibung der Funktionalität*“)
- SE 2.1 System technisch entwerfen
  („Schnittstellenübersicht.*Systemexterne Schnittstellen*“,
  „Schnittstellenübersicht.*Systeminterne Schnittstellen*“)
- SE 3.2 Anforderungen an die externen Schnittstellen der SW-/HW-Einheit präzisieren
  („Technische Anforderungen.*Technische Anforderungen an die Schnittstellen*“)
- SE 3.3 Anforderungen an die Funktionalität definieren
  („Technische Anforderungen.*Gesamtfunktion*“)

Methode

- Datenfluß-Modellierung (DFM)
- Kontrollfluß-Modellierung (KFM)

3.6.7.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen

- Datenflüsse zwischen Objekten (Prozessen, Datenspeichern, Terminatoren) modelliert und
- diese Datenflüsse um Steuerungselemente im Sinne der Kontrollfluß-Modellierung ergänzt werden können.

3.6.7.3 Anforderungen

3.6.7.3.1 Anforderungen an die Schnittstellen

LSE07.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE07.S.2 Eingangs-Schnittstelle zu LSE05 „Funktionsstrukturierung unterstützen“
Mit LSE05 erfaßte Funktionen aus Funktionsstrukturen werden ohne weitere Transformation aus der Objektverwaltung zur automatischen Einbindung in Datenflußdiagramme übernommen.

Die in LSE05 definierten Funktionen entsprechen den in Datenflußdiagrammen verwendeten Prozessen.

LSE07.S.3 Ausgangs-Schnittstelle zu LSE09 „Informationsstrukturierung unterstützen“
Es ist möglich, Prozesse, Datenflüsse und Datenspeicher aus Datenflußdiagrammen über die Objektverwaltung an LSE09 als Bezugspunkte für die Definition von Informationsstrukturen zu übergeben.

Die in LSE05 definierten Funktionen entsprechen den in Datenflußdiagrammen verwendeten Prozessen.

LSE07.S.4 Ausgangs-Schnittstelle zu LQS14 „Prüffälle generieren“
Es ist möglich, Datenflußdiagramme und Kontrollflußdiagramme ohne weitere Transformation über die Objektverwaltung an LQS14 zur Generierung von Prüffällen zu übergeben.

3.6.7.3.2 Anforderungen an die Methodenunterstützung

LSE07.M.1 Datenfluß-Modellierung (DFM)

LSE07.M.1.1 Symbole

LSE07.M.1.1.1 Prozesse
Es stehen Symbole zur Darstellung von Prozessen zur Verfügung.

LSE07.M.1.1.2 Terminatoren
Es stehen Symbole zur Darstellung von Terminatoren zur Verfügung.

LSE07.M.1.1.3 Datenflüsse
Es stehen Symbole zur Darstellung von Datenflüssen zur Verfügung.

LSE07.M.1.1.4 Datenspeicher
Es stehen Symbole zur Darstellung von Datenspeichern zur Verfügung.

LSE07.M.1.2 Datenflüsse

LSE07.M.1.2.1 Terminatoren und Datenspeicher
Es ist sichergestellt, daß keine Datenflüsse zwischen Terminatoren und Datenspeichern angelegt werden.

LSE07.M.1.2.2 Datenflüsse zwischen Terminatoren
Es ist sichergestellt, daß keine Datenflüsse zwischen Terminatoren angelegt werden.

LSE07.M.1.2.3 Auslöser
Es ist möglich, bei jedem Datenfluß einen Auslöser zu dokumentieren.

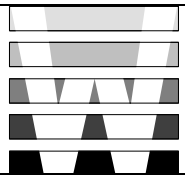
LSE07.M.1.2.4 Aufspalten bzw. Zusammenführen
Es ist möglich, Datenflüsse aufzuspalten oder zusammenzuführen.

LSE07.M.1.2.5 Umdrehen
Es ist möglich, die Richtung eines Datenflusses umzudrehen.

LSE07.M.1.2.6 Umleiten
Es ist möglich, Datenflüsse umzuleiten.

LSE07.M.1.3 Hierarchie

LSE07.M.1.3.1 Hierarchische Verfeinerung
Es ist möglich, die in den Datenflußdiagrammen verwendeten Prozesse hierarchisch zu verfeinern.



LSE07.M.1.3.2 Automatisches Anzeigen
Ein- und ausgehende Datenflüsse eines Prozesses werden auf der nächst tieferen Hierarchieebene automatisch angezeigt.

LSE07.M.1.4 Öffnen von Datenflußdiagrammen

LSE07.M.1.4.1 Nächst tiefere Ebene
Es ist möglich, aus einem Datenflußdiagramm ein Datenflußdiagramm der nächst tieferen Ebene innerhalb der Baumstruktur zu öffnen.

LSE07.M.1.4.2 Nächst höhere Ebene
Es ist möglich, aus einem Datenflußdiagramm ein Datenflußdiagramm der nächst höheren Ebene innerhalb der Baumstruktur zu öffnen.

LSE07.M.2 Kontrollfluß-Modellierung (KFM)

LSE07.M.2.1 Symbole

LSE07.M.2.1.1 Kontrollflüsse
Es stehen Symbole zur Darstellung der Kontrollspeicher zur Verfügung.

LSE07.M.2.1.2 Kontrollspeicher
Es stehen Symbole zur Darstellung der Kontrollflüsse zur Verfügung.

LSE07.M.2.1.3 CSPEC-Schnittstelle
Es steht je Kontrollflußdiagramm ein Symbol für die Schnittstelle zur Kontrollspezifikation (CSPEC-Schnittstelle) zur Verfügung. Hierin kann dargestellt werden, ob ein Prozeß in Abhängigkeit von CSPEC aktiviert oder deaktiviert wird und ob ein aktivierter Prozeß als Eingabegröße in der CSPEC wirkt.

LSE07.M.2.1.4 Prozesse
Es stehen Symbole zur Darstellung von Prozessen zur Verfügung.

LSE07.M.2.1.5 Terminatoren
Es stehen Symbole zur Darstellung von Terminatoren zur Verfügung.

Mit obigen Symbolen läßt sich ein Kontrollflußdiagramm analog einem Datenflußdiagramm aufbauen. Ein Kontrollflußdiagramm zeigt lediglich Kontrollflüsse anstelle von Datenflüssen. Die Prozesse eines Kontrollflußdiagramms symbolisieren jedoch keine Verarbeitung der eintreffenden Kontrollflüsse. Sie zwingen lediglich den Kontrollfluß, derselben Verfeinerungsstruktur zu folgen wie der Datenfluß.

LSE07.M.2.2 Hierarchie

LSE07.M.2.2.1 Hierarchische Verfeinerung
Es ist möglich, die in den Kontrollflußdiagrammen verwendeten Prozesse hierarchisch zu verfeinern.

LSE07.M.2.2.2 Automatische Anzeige
Ein- und ausgehende Kontrollflüsse eines Prozesses werden auf der nächst tieferen Hierarchieebene automatisch angezeigt.

LSE07.M.2.3 Öffnen von Kontrolllflußdiagrammen

LSE07.M.2.3.1 Nächst tiefere Ebene
Es ist möglich, aus einem Kontrollflußdiagramm ein Kontrollflußdiagramm der nächst tieferen Ebene innerhalb der Baumstruktur zu öffnen.

LSE07.M.2.3.2 Nächst höhere Ebene
Es ist möglich, aus einem Kontrollflußdiagramm ein Kontrollflußdiagramm der nächst höheren Ebene innerhalb der Baumstruktur zu öffnen.

3.6.7.3.3 Anforderungen an die Funktionen

keine

3.6.7.3.4 Sonstige Anforderungen

LSE07.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE07.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE07.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.8 Leistungseinheit LSE08 „Modellierung der Funktionsdynamik unterstützen“

3.6.8.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 1.5 System fachlich strukturieren
(„Anwenderforderungen.*Beschreibung der Funktionalität*“)

– SE 3.3 Anforderungen an die Funktionalität definieren
(„Technische Anforderungen.*Gesamtfunktion*“)

Methode

– Funktionsnetz-Modellierung (FNET)

– Systemverhaltensmodelle (SVM)

3.6.8.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Modellierung dynamischer Aspekte verschiedener Funktionen einer Anwendung über mehrere Detaillierungsebenen hinweg gestellt. Es wird die zeitliche Abfolge von Funktionen in Abläufen beschrieben, um das Verhalten von Anwendungen zu beobachten, die nicht rein sequentiell arbeiten.

3.6.8.3 Anforderungen

3.6.8.3.1 Anforderungen an die Schnittstellen

LSE08.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE08.S.2 Eingangs-Schnittstelle zu LSE05 „Funktionsstrukturierung unterstützen“
Mit LSE05 erfaßte Funktionen aus Funktionsstrukturen werden ohne weitere Transformation aus der Objektverwaltung zur automatischen Einbindung in Ablaufmodelle übernommen.

Die in LSE05 definierten Funktionen entsprechen den in Ablaufmodellen verwendeten Funktionen.

LSE08.S.3 Ausgangs-Schnittstelle zu LQS14 „Prüffälle generieren“
Es ist möglich, Ablaufmodelle ohne weitere Transformation über die Objektverwaltung an LQS14 zur Generierung von Prüffällen zu übergeben.

3.6.8.3.2 Anforderungen an die Methodenunterstützung

LSE08.M.1 Funktionsnetz-Modellierung (FNET)

LSE08.M.1.1 Symbole

LSE08.M.1.1.1 Funktionen
Es stehen Symbole zur Darstellung von Funktionen zur Verfügung.

LSE08.M.1.1.2 Bedingungen
Es stehen Symbole zur Darstellung von Bedingungen zur Verfügung.

LSE08.M.1.1.3 Auslöser
Es stehen Symbole zur Darstellung von Auslösern zur Verfügung.

LSE08.M.1.1.4 Verteiler
Es stehen Symbole zur Darstellung von Verteilern zur Verfügung.

LSE08.M.1.1.5 Ablaufpfeile
Es stehen Symbole zur Darstellung von Ablaufpfeilen zur Verfügung.

LSE08.M.1.1.6 Marken
Es stehen Symbole zur Darstellung von Marken zur Verfügung.

LSE08.M.1.2 Beschreiben von Bedingungen

LSE08.M.1.2.1 Datenbedingungen
Es ist möglich, Bedingungen als Datenbedingungen zu kennzeichnen.

LSE08.M.1.2.2 Zeitbedingungen
Es ist möglich, Bedingungen als Zeitbedingungen zu kennzeichnen.

LSE08.M.1.2.3 Zustandsbedingungen
Es ist möglich, Bedingungen als Zustandsbedingungen zu kennzeichnen.

Zustandsbedingungen können Datenbedingungen sein, aber auch Zustandsveränderungen in einem Informationssystem beschreiben.

LSE08.M.1.3 Logische Ausdrücke für Auslöser
Es ist möglich, Auslöser durch logische Ausdrücke zu beschreiben.

LSE08.M.1.4 Logische Ausdrücke für Verteiler
Es ist möglich, Verteiler durch logische Ausdrücke zu beschreiben.

LSE08.M.1.5 Auslöser für Funktionen
Es ist sichergestellt, daß jede Funktion genau einen Auslöser besitzt.

LSE08.M.1.6 Verteiler von Funktionen
Es ist sichergestellt, daß jede Funktion mindestens einen Verteiler besitzt.

LSE08.M.1.7 Beginn von Abläufen
Es ist sichergestellt, daß jeder Ablauf immer mit einer Daten- oder Zeitbedingung beginnt.

LSE08.M.1.8 Ende von Abläufen
Es ist sichergestellt, daß jeder Ablauf immer mit einer Datenbedingung endet.

LSE08.M.2 Systemverhaltensmodelle (SVM)

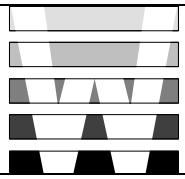
LSE08.M.2.1 Petri-Netze

LSE08.M.2.1.1 Symbole

LSE08.M.2.1.1.1 Symbole für Instanzen
Es steht ein Symbol zur Darstellung von Instanzen (Ereignisse, Transitionen) zur Verfügung.

LSE08.M.2.1.1.2 Symbole für Kanäle
Es stehen Symbole zur Darstellung von Kanälen (Bedingungen, Stellen, Prädikate) zur Verfügung.

LSE08.M.2.1.1.3 Symbole für Kanten
Es stehen Symbole zur Darstellung von Kanten zwischen Instanzen und Kanälen zur Verfügung.



LSE08.M.2.1.1.4 Symbole für Marken
Es stehen Symbole zur Darstellung von Marken zur Verfügung.

LSE08.M.2.1.2 Kapazitäten
Es ist möglich, Kanälen Kapazitäten zuzuordnen.

Die Kapazität bestimmt die Anzahl der maximal von einem Kanal zu einem Zeitpunkt aufnehmbaren Marken.

LSE08.M.2.1.3 Gewichtung
Es ist möglich, Kanten Gewichte zuzuordnen.

Das Gewicht gibt an, wieviele Marken bei einem Zustandsübergang aus einem Kanal entnommen bzw. in einem Kanal abgelegt werden können.

LSE08.M.2.1.4 Anfangsbelegung
Es ist möglich, Kanälen eine Anfangsbelegung zuzuordnen.

LSE08.M.2.1.5 Schaltregeln
Es ist möglich, semantische Festlegungen für das Schalten einer Instanz zu treffen.

Die Schaltregeln ergeben sich entweder aus der Art des betrachteten Netzes (Bedingungs/Ereignis-Netz, Stellen/Transitions-Netz) oder werden in Form von programmiersprachlicher Beschreibung angegeben (Prädikat/Transitions-Netz).

LSE08.M.2.1.6 Zeitbedingungen
Es ist möglich, Instanzen mit Zeitbedingungen zu versehen.

LSE08.M.2.1.7 Hierarchische Netze

LSE08.M.2.1.7.1 Verfeinerung
Es ist möglich, Netze in einer Hierarchie top down zu verfeinern.

LSE08.M.2.1.7.2 Vergröberung
Es ist möglich, Netze in einer Hierarchie bottom up zu vergröbern.

LSE08.M.2.1.8 Fusion
Es ist möglich, eine im Netz nur einmal existierende Komponente (Kanal, Instanz) in mehrfachen grafischen Ausprägungen zu verwenden.

LSE08.M.2.2 Statecharts

LSE08.M.2.2.1 Symbole

LSE08.M.2.2.1.1 Symbole für Zustände
Es steht ein Symbol zur Darstellung von Zuständen zur Verfügung.

LSE08.M.2.2.1.2 Symbole für Ereignisse
Es steht ein Symbol zur Darstellung von Transitionen zur Verfügung.

LSE08.M.2.2.2 Ereignisse
Es ist möglich, Transitionen Ereignisse zuzuordnen.

LSE08.M.2.2.3 Attribute
Es ist möglich, Ereignisse durch Attribute zu beschreiben.

LSE08.M.2.2.4 Ereignisklassen
Es ist möglich, Ereignisklassen zu definieren.

LSE08.M.2.2.5 Bedingungen
Es ist möglich, Transitionen Bedingungen (logische Ausdrücke) zuzuordnen.

Bedingungen kontrollieren Transitionen. Ein Zustandsübergang kann nur stattfinden, wenn neben dem eintretenden Ereignis auch alle Bedingungen erfüllt sind.

LSE08.M.2.2.6 Aktionen
Es ist möglich, Transitionen Aktionen zuzuordnen.

Aktionen sind an Ereignisse gebunden. Eine Aktion repräsentiert eine Operation, deren Dauer gegenüber der Zeitbasis des Statecharts nicht signifikant ist (Beispiel: Anzeige eines Menüs bei Betätigen einer Maustaste).

LSE08.M.2.2.7 Aktivitäten
Es ist möglich, Zuständen Aktivitäten zuzuordnen.

Aktionen sind Operationen, deren Ausführung Zeit benötigt. Eine Aktivität beginnt mit dem Eintreten eines Zustands und terminiert entweder normal oder durch Eintreten eines Ereignisses, das eine Zustandsveränderung bewirkt.

LSE08.M.2.2.8 Verknüpfung von Statecharts
Es ist möglich, Statecharts über gemeinsame Ereignisse zu verbinden.

LSE08.M.2.2.9 Hierarchische Statecharts

LSE08.M.2.2.9.1 Verfeinerung
Es ist möglich, die Zustände eines Statecharts in einer Hierarchie top down zu verfeinern.

LSE08.M.2.2.9.2 Vergröberung
Es ist möglich, Statecharts in einer Hierarchie bottom up zu vergröbern.

LSE08.M.2.3 Specification and Description Language (SDL)

LSE08.M.2.3.1 Symbole

LSE08.M.2.3.1.1 Symbole für Block-Interaktionsdiagramme
Es stehen Symbole für Umgebung, Block, Prozeß, Signalliste, Kanal, Signalweg, Create zur Verfügung.

LSE08.M.2.3.1.2 Symbole für Prozeßdiagramme
Es stehen Symbole für Start, Stop, State, Nextstate, Input, Output, Task, Create Request, Flußlinie, Entscheidung, Save, Definition und Aufruf von Prozeduren und Makros, Eingangs- und Ausgangskonnektor, Kommentar, Texterweiterung zur Verfügung.

LSE08.M.2.3.2 Notationen

LSE08.M.2.3.2.1 Textuelle Notationen
In Verbindung mit den grafischen Symbolen gibt es an syntaktisch zulässigen Stellen die Möglichkeit für semantisch bedeutsame textuelle Notationen.

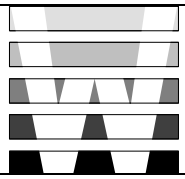
LSE08.M.2.3.2.2 Notationen für weitere SDL-Entitäten
Es gibt textuelle Notationen für Signalaufzählung, Signaldefinition, formale Parameter, gültige Eingangssignalmenge, Variablendefinition, View Definition, Viewing Expression in textueller SDL-Syntax.

LSE08.M.2.3.2.3 Namen und Identifikatoren
Es gibt textuelle Notationen für Namen und Identifikatoren (= Name und Qualifier) in textueller SDL-Syntax zur eindeutigen Bezeichnung aller SDL-Entitäten, wo dies semantisch notwendig ist.

LSE08.M.2.3.3 Regeln

LSE08.M.2.3.3.1 Statische Regeln
Es ist nicht möglich, Entitäten entgegen der statischen Verknüpfungsregeln von SDL miteinander zu verbinden

LSE08.M.2.3.3.2 Dynamische Regeln
Im Falle der Ausführung einer gegebenen Systembeschreibung wird die dynami-



sche Semantik der Signalabarbeitung befolgt (Signalspeicherung in der Prozeßeingabewarteschlange, Save-Mechanismus, Nullübergänge).

LSE08.M.2.3.4 Strukturierungseigenschaften

LSE08.M.2.3.4.1 Hierarchische Verfeinerung
Es ist möglich, Blöcke und Prozesse beliebig tief hierarchisch zu verfeinern.

LSE08.M.2.3.4.2 Block-Baumdiagramm
Es ist möglich, einen einzelnen Block in Sub-Blöcke zu zerlegen und mit diesen zusammen ein Block-Baumdiagramm zu erstellen. Auch die gesamte Zerlegung eines Systems in Blöcke und Sub-Blöcke kann, beginnend mit dem Block „System“, als ein Block-Baumdiagramm beliebiger Tiefe dargestellt werden.

LSE08.M.2.3.4.3 Prozeß-Baumdiagramm
Es ist möglich, einen einzelnen Prozeß in Sub-Prozesse zu zerlegen und mit diesen zusammen ein Prozeß-Baumdiagramm zu erstellen. Auch die gesamte Prozeßzerlegung eines Systems kann als ein Prozeß-Baumdiagramm beliebiger Tiefe dargestellt werden.

LSE08.M.2.3.4.4 Prozeß-Block-Zuordnung
Es ist möglich, Prozesse oder Sub-Prozesse einzelnen Blöcken oder Sub-Blöcken durch Symbole eindeutig zuzuordnen.

3.6.8.3.3 Anforderungen an die Funktionen

keine

3.6.8.3.4 Sonstige Anforderungen

LSE08.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE08.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE08.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.9 Leistungseinheit LSE09 „Informationsstrukturierung unterstützen“

3.6.9.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- SE 1.1 Ist-Aufnahme/-Analyse durchführen („Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“)
- SE 1.5 System fachlich strukturieren („Anwenderforderungen.*Beschreibung der Funktionalität*“)
- SE 2.4 Anwenderforderungen zuordnen („Systemarchitektur.*Anforderungszuordnung*“)
- SE 3.3 Anforderungen an die Funktionalität definieren („Technische Anforderungen.*Gesamtfunktion*“)
- SE 5.1-SW SW-Komponente/-Modul/Datenbank beschreiben („SW-Entwurf.*Datenbankbeschreibung*“ und „Datenkatalog“)

Methode

- E/R-Modellierung (ER)
- Entity Life History (ELH)
- Normalisierung (NORM)

3.6.9.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Unterstützung der Methoden

- „Entity-Relationship-Modellierung“ (E/R-Modellierung),
- „Entity Life History“ (ELH) und
- „Normalisierung“ (NORM)

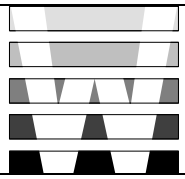
gestellt.

3.6.9.3 Anforderungen

3.6.9.3.1 Anforderungen an die Schnittstellen

LSE09.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE09.S.2 Eingangs-Schnittstelle zu LSE05 „Funktionsstrukturierung unterstützen“
Es ist möglich, mit LSE05 erfaßte Funktionen aus der Objektverwaltung gezielt als Bezugspunkte für die Erstellung von Informationsstrukturen auszuwählen.



LSE09.S.3 Eingangs-Schnittstelle zu LSE07 „Modellierung der Informationsflüsse unterstützen“
Es ist möglich, mit LSE07 erfaßte Prozesse, Datenflüsse und Datenspeicher aus Datenflußdiagrammen aus der Objektverwaltung gezielt als Bezugspunkte für die Erstellung von Informationsstrukturen auszuwählen.

LSE09.S.4 Eingangs-Schnittstelle zu LSE21 „Datenbanken, Datenstrukturen rückwärts transformieren“
Es ist möglich, Informationsstrukturen, die mit LSE21 aus logischen Schemaspezifikationen für Datenbanken und Datenstrukturen erstellt wurden, ohne weitere Transformation aus der Objektverwaltung zu übernehmen.

LSE09.S.5 Ausgangs-Schnittstelle zu LSE11 „Spezifikation der Datenbanken unterstützen“
Es ist möglich, Informationsstrukturen ohne weitere Transformation über die Objektverwaltung an LSE11 zur Umsetzung in logische Schemaspezifikationen für Datenbanken und Datenstrukturen zu übergeben.

LSE09.S.6 Ausgangs-Schnittstelle zu LSE21 „Datenbanken, Datenstrukturen rückwärts transformieren“
Es ist möglich, Informationsstrukturen ohne weitere Transformation über die Objektverwaltung an LSE21 zur Berücksichtigung bei der Analyse von Beschreibungen zu Datenbanken und Datenstrukturen zu übergeben.

3.6.9.3.2 Anforderungen an die Methodenunterstützung

LSE09.M.1 Entity-Relationship-Modellierung (E/R-Modellierung)

LSE09.M.1.1 Symbole

LSE09.M.1.1.1 Fundamentale Entitätstypen
Es stehen Symbole zur Darstellung von fundamentalen Entitätstypen zur Verfügung.

LSE09.M.1.1.2 Attributive Entitätstypen
Es stehen Symbole zur Darstellung von Entitätstypen zur Verfügung, die der Auslagerung von Wiederholgruppen dienen.

LSE09.M.1.1.3 Assoziative Entitätstypen
Es stehen Symbole zur Darstellung von Entitätstypen zur Verfügung, die der Auflösung von N:M-Beziehungen dienen.

LSE09.M.1.1.4 Beziehungstypen und Kardinalitäten
Es stehen Symbole zur Darstellung von Beziehungstypen zur Verfügung.

LSE09.M.1.1.5 Aggregationen
Es stehen Symbole zur Darstellung von Aggregationen zur Verfügung,

Aggregationen können z. B. durch eine Beziehung vom Typ „Ist-Teil-von“ dargestellt werden. Entitätstypen, die durch einen solchen Beziehungstyp zu einem übergeordneten Entitätstyp zusammengefaßt werden, sind als Entitätssubtypen zu dem entsprechenden Entitätssupertyp aufzufassen.

LSE09.M.1.1.6 Spezialisierungen
Es stehen Symbole zur Darstellung von Spezialisierungen zur Verfügung,

Spezialisierungen können z. B. durch eine Beziehung vom Typ „Ist-ein“ dargestellt werden. Entitätstypen, die durch einen solchen Beziehungstyp einem Entitätstyp als Spezialfall untergeordnet werden, sind als Entitätssubtypen zu dem entsprechenden Entitätssupertyp aufzufassen.

LSE09.M.1.2 Bearbeiten von Beziehungstypen

LSE09.M.1.2.1 Umhängen von Beziehungstypen
Es ist möglich, Beziehungstypen umzuhängen.

LSE09.M.1.2.2 Ändern der Beziehungsrichtung
Es ist möglich, die Beziehungsrichtung von Beziehungstypen zu ändern.

LSE09.M.1.2.3 Beziehungstypen immer zwischen Entitätstypen
Es ist sichergestellt, daß nur Beziehungstypen angelegt werden, die Entitätstypen miteinander verbinden.

LSE09.M.1.2.4 Kardinalitäten
Es ist sichergestellt, daß nur Beziehungstypen angelegt werden, denen in den E/R-Diagrammen erkennbare Kardinalitäten zugeordnet sind.

LSE09.M.1.3 Beschreiben von Entitätstypen mit Attributen

LSE09.M.1.3.1 Attribute
Es ist möglich, Entitätstypen durch Attribute zu beschreiben.

LSE09.M.1.3.2 Beziehungstypen als Attribut
Es ist möglich, Beziehungstypen als Attribute zu verwenden.

LSE09.M.1.3.3 Vererbung
Es ist möglich, Attribute an Entitätstypen zu vererben.

Eine Vererbung von Attributen ist bei der Subtypbildung relevant.

LSE09.M.1.3.4 Gruppe von Attributen
Es ist möglich, Attribute zu einer Gruppe zusammenzufassen.

LSE09.M.1.3.5 Umbenennung von Attributen
Es ist möglich, Attribute – bis auf (Teil-) Schlüsselattribute – umzubenennen.

LSE09.M.1.3.6 Löschung von Attributen
Es ist möglich, Attribute zu löschen.

LSE09.M.1.3.7 Zurückweisen bereits verwendeter Attributnamen
Beim Hinzufügen von Attributen werden bereits verwendete Attributnamen zurückgewiesen.

LSE09.M.1.3.8 Primärschlüssel
Es ist möglich, Attribute oder Gruppen von Attributen als Primärschlüssel auszuzeichnen.

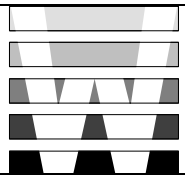
LSE09.M.1.3.9 Alternative Primärschlüssel
Es ist möglich, Attribute oder Gruppen von Attributen als alternative Primärschlüssel auszuzeichnen.

LSE09.M.1.3.10 Datentyp
Es ist möglich, Attributen einen Datentyp zuzuordnen.

LSE09.M.1.3.11 Wertebereich
Es ist möglich, Attributen Wertebereiche zuzuordnen.

LSE09.M.1.3.12 Löschen eines Entitätstyps
Es ist sichergestellt, daß vor dem Löschen eines Entitätstyps dessen Verwendungen in Aggregationen und Spezialisierungen angezeigt werden.

LSE09.M.1.3.13 Automatisches Löschen von Attributen
Es ist sichergestellt, daß mit dem Löschen eines Entitätstyps automatisch alle zugehörigen Attribute gelöscht werden.



LSE09.M.1.3.14 Gesamtmodell
Es ist möglich, ein Gesamtmodell aus in verschiedenen E/R-Modellen angelegten Entitäts- und Beziehungstypen zu erstellen

LSE09.M.1.3.15 Teildatenmodell
Es ist möglich, Teildatenmodelle aus einem bestehenden Gesamtdatendmodell zu erstellen.

LSE09.M.1.3.16 Konsolidierungen
Es ist möglich, ein Teildatenmodell in ein Gesamtmodell einzufügen, wobei die Integration erst nach Beseitigung aller Inkonsistenzen erfolgt.

LSE09.M.2 Entity Life History (ELH)

LSE09.M.2.1 Symbole

LSE09.M.2.1.1 Zustandssymbol
Es steht ein Symbol zur Darstellung des Zustands eines Entitätstyps zur Verfügung, das mit dem Namen des Zustandes beschriftet werden kann.

LSE09.M.2.1.2 Übergangssymbol
Es steht ein Symbol zur Darstellung des Übergangs von einem Zustand zu einem anderen Zustand zur Verfügung, das mit dem Namen des zustandsändernden Ereignisses beschriftet werden kann.

LSE09.M.2.2 Zustandsdiagramm
Es ist möglich, für jeden Entitätstyp automatisch ein Zustandsdiagramm zu erzeugen.

LSE09.M.2.3 Zustandsänderungen
Es ist möglich, im Zustandsdiagramm alle Zustände, sowie die zustandsändernden Ereignisse im Lebenslauf der zugeordneten Entität, darzustellen.

LSE09.M.2.4 Integritätsbedingungen
Es ist möglich, einer Entität alle Integritätsbedingungen zuzuordnen, die zu deren zustandsändernden Ereignissen im Lebenslauf gehören.

LSE09.M.3 Normalisierung (NORM)

LSE09.M.3.1 Schlüsselattribute
Es ist möglich, in einer Relation Schlüsselattribute zu kennzeichnen.

LSE09.M.3.2 Funktional abhängige Attribute
Es ist möglich, in einer Relation von Schlüssel- und Teilschlüsselattributen funktional abhängige Attribute zu kennzeichnen.

LSE09.M.3.3 Transitiv abhängige Attribute
Es ist möglich, nicht zum Schlüssel gehörige voneinander transitiv abhängige Attribute zu kennzeichnen.

LSE09.M.3.4 Erste Normalform
Es ist möglich, eine unnormalisierte Relation automatisch in Relationen in erster Normalform zu überführen, wenn in der unnormalisierten Relation der Schlüssel sowie alle vom Schlüssel nicht funktional abhängigen Attribute gekennzeichnet sind.

LSE09.M.3.5 Zweite Normalform
Es ist möglich, eine Relation in erster Normalform automatisch in Relationen in zweiter Normalform zu überführen, wenn in der Relation in erster Normalform neben dem Schlüssel jedes nicht dem Schlüssel angehörige Attribut, das von einem Schlüsselteil funktional abhängig ist, gekennzeichnet ist.

LSE09.M.3.6 Dritte Normalform
Es ist möglich, eine Relation in zweiter Normalform automatisch in Relationen in dritter Normalform zu überführen, wenn in der Relation in zweiter Normalform neben dem Schlüssel alle nicht zum Schlüssel gehörigen voneinander transitiv abhängigen Attribute gekennzeichnet werden.

LSE09.M.3.7 Denormalisierung
Es ist möglich, alle Normalisierungen einer Relation automatisch rückgängig zu machen.

3.6.9.3.3 Anforderungen an die Funktionen

keine

3.6.9.3.4 Sonstige Anforderungen

LSE09.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE09.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE09.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.10 Leistungseinheit LSE10 „Spezifikation der Komponenten und Module unterstützen“

3.6.10.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 5.1-SW SW-Komponente/-Modul/Datenbank beschreiben
(„SW-Entwurf.*SW-Komponenten-/SW-Modul-Beschreibung*“)

Methode

– Pseudocode (PCODE)

3.6.10.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Spezifikation der Ablauflogik und Datenbehandlung für ein SW-Modul bzw. eine SW-Komponente gestellt.

3.6.10.3 Anforderungen

3.6.10.3.1 Anforderungen an die Schnittstellen

LSE10.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE10.S.2 Eingangs-Schnittstelle zu LSE02 „Spezifikation von Nutzerschnittstellen unterstützen“
Es ist möglich, mit LSE02 erstellte Spezifikationen für Schnittstellen ohne weitere Transformation über die Objektverwaltung zur Einbindung in die Spezifikation von Komponenten und Modulen zu übernehmen.

LSE10.S.3 Eingangs-Schnittstelle zu LSE03 „Architekturentwurf unterstützen“
Es ist möglich, mit LSE03 in einer Architektur festgelegte Komponenten und Module aus der Objektverwaltung gezielt als Bezugspunkte für die Spezifikation auszuwählen.

LSE10.S.4 Eingangs-Schnittstelle zu LSE06 „Spezifikation der Funktionen unterstützen“
Es ist möglich, mit LSE06 erstellte Spezifikationen für Funktionen ohne weitere Transformation über die Objektverwaltung gezielt für die Spezifikation von Komponenten und Modulen zu übernehmen.

LSE10.S.5 Eingangs-Schnittstelle zu LSE11 „Spezifikation der Datenbanken unterstützen“
Es ist möglich, mit LSE11 erfaßte logische Schemaspezifikationen für Datenbanken und Datenstrukturen ohne weitere Transformation über die Objektverwaltung zur Einbindung in die Spezifikation von Komponenten und Modulen zu übernehmen.

LSE10.S.6 Eingangs-Schnittstelle zu LSE20 „Code rückwärts transformieren“
Es ist möglich, Spezifikationen für Komponenten und Module, die mit LSE20 aus dem Quellcode erzeugt wurden, ohne weitere Transformation aus der Objektverwaltung zu übernehmen.

LSE10.S.7 Ausgangs-Schnittstelle zu LSE12 „Komponenten und Module generieren“
Es ist möglich, Spezifikationen für Komponenten und Module ohne weitere Transformation über die Objektverwaltung an LSE12 zur Generierung von Quellcode zu übergeben.

LSE10.S.8 Ausgangs-Schnittstelle zu LQS14 „Prüffälle generieren“
Es ist möglich, Spezifikationen für Komponenten und Module ohne weitere Transformation über die Objektverwaltung an LQS14 zur Generierung von Prüffällen zu übergeben.

3.6.10.3.2 Anforderungen an die Methodenunterstützung

LSE10.M.1 Pseudocode (PCODE)

LSE10.M.1.1 Symbole

LSE10.M.1.1.1 Kontrollstrukturen
Es stehen Symbole zur Darstellung der Kontrollstrukturen Sequenz, Selektion und Iteration zur Verfügung.

LSE10.M.1.1.2 Beginn und Ende
Es stehen Symbole zur Darstellung von Beginn und Ende einer Kontrollstruktur zur Verfügung.

LSE10.M.1.1.3 Verarbeitungsabbruch
Es stehen Symbole zur Darstellung eines Abbruchs der Verarbeitung innerhalb einer Kontrollstruktur zur Verfügung (Abbruchanweisung).

LSE10.M.1.1.4 Anweisungen
Es stehen Symbole zur Darstellung von programmiersprachlichen Anweisungen zur Verfügung.

Unter programmiersprachlichen Anweisungen sind hier vor allem Zuweisungen, Berechnungen und Vergleiche aufzufassen.

LSE10.M.1.1.5 Datendeklarationen
Es stehen Symbole zur Darstellung von Datendeklarationen bis auf die Ebene einzelner Datenelemente zur Verfügung.

LSE10.M.1.1.6 Datenzugriffe
Es stehen Symbole zur Darstellung von Datenzugriffen zur Verfügung.

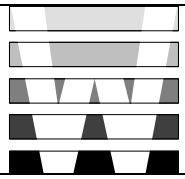
LSE10.M.1.1.7 Parameter
Es stehen Symbole zur Darstellung von Parametern zur Verfügung.

LSE10.M.1.1.8 Masken
Es stehen Symbole zur Darstellung von Maskenaufrufen zur Verfügung.

LSE10.M.1.1.9 Komponenten-/Modulaufrufe
Es stehen Symbole zur Darstellung von Komponenten- und Modulaufrufen zur Verfügung.

LSE10.M.1.1.10 Verarbeitungsende
Es stehen Symbole zur Darstellung einer Beendigung der Verarbeitung innerhalb einer Komponente bzw. eines Moduls zur Verfügung.

LSE10.M.1.1.11 Kommentare
Es stehen Symbole zur Darstellung von Kommentaren zur Verfügung.



LSE10.M.1.1.12 Zielsprache einstellen
Es ist möglich, die Syntax der verschiedenen Konstrukte abhängig von der jeweiligen Zielumgebung einzustellen.

LSE10.M.1.1.13 Verwendung von Sprachelementen
Es wird sichergestellt, daß keine Sprachelemente verwendet werden, die in der Zielsprache keine Entsprechung haben.

3.6.10.3.3 Anforderungen an die Funktionen

keine

3.6.10.3.4 Sonstige Anforderungen

LSE10.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE10.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE10.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.11 Leistungseinheit LSE11 „Spezifikation der Datenbanken unterstützen“

3.6.11.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 5.1-SW SW-Komponente/-Modul/Datenbank beschreiben („SW-Entwurf.*Datenbankbeschreibung*“ und „Datenkatalog“)

Methode

– Data Navigation Modelling (DNAV)
– Logische DB-Modellierung (LOGM)

3.6.11.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen

– Parameter zum Mengengerüst, den geplanten Zugriffsfolgen, -arten und -häufigkeiten, Prioritäten der Verarbeitung und dem Zeitverhalten – insbesondere dem geforderten Antwortzeitverhalten – erfaßt werden
– konzeptionelle Schemaspezifikationen in zugehörige logische Schemaspezifikationen transformiert werden können.

3.6.11.3 Anforderungen

3.6.11.3.1 Anforderungen an die Schnittstellen

LSE11.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE11.S.2 Eingangs-Schnittstelle zu LSE03 „Architekturentwurf unterstützen“
Es ist möglich, mit LSE03 in einer Architektur festgelegte Datenbanken aus der Objektverwaltung gezielt als Bezugspunkte für die Spezifikation auszuwählen.

LSE11.S.3 Eingangs-Schnittstelle zu LSE09 „Informationsstrukturierung unterstützen“
Es ist möglich, mit LSE09 erstellte Informationsstrukturen ohne weitere Transformation aus der Objektverwaltung zur Umsetzung in logische Schemaspezifikationen für Datenbanken zu übernehmen.

LSE11.S.4 Eingangs-Schnittstelle zu LSE21 „Datenbanken, Datenstrukturen rückwärts transformieren“
Es ist möglich, logische Schemaspezifikationen für Datenbanken und Datenstrukturen, die mit LSE21 aus physischen Schemaspezifikationen erzeugt wurden, ohne weitere Transformation aus der Objektverwaltung zu übernehmen.

LSE11.S.5 Ausgangs-Schnittstelle zu LSE10 „Spezifikation der Komponenten und Module unterstützen“
Es ist möglich, logische Schemaspezifikationen für Datenbanken und Datenstrukturen ohne weitere Transformation über die Objektverwaltung an LSE10 zur Einbindung in die Spezifikation von Komponenten und Modulen zu übergeben.

LSE11.S.6 Ausgangs-Schnittstelle zu LSE13 „Datenbanken generieren“
Es ist möglich, logische Schemaspezifikationen für Datenbanken und Datenstrukturen ohne weitere Transformation über die Objektverwaltung an LSE13 zur Generierung von physischen Schemaspezifikationen zu übergeben.

LSE11.S.7 Ausgangs-Schnittstelle zu LSE19 „Zeitverhalten simulieren“
Es ist möglich, logische Schemaspezifikationen für Datenbanken und Datenstrukturen ohne weitere Transformation über die Objektverwaltung an LSE19 zur Simulation des voraussichtlichen Zeitverhaltens zu übergeben.

LSE11.S.8 Ausgangs-Schnittstelle zu LSE21 „Datenbanken, Datenstrukturen rückwärts transformieren“
Es ist möglich, logische Schemaspezifikationen für Datenbanken und Datenstrukturen ohne weitere Transformation über die Objektverwaltung an LSE21 zur Umsetzung in Informationsstrukturen zu übergeben.

LSE11.S.9 Ausgangs-Schnittstelle zu LQS14 „Prüffälle generieren“
Es ist möglich, logische Schemaspezifikationen für Datenbanken und Datenstrukturen ohne weitere Transformation über die Objektverwaltung an LQS14 zur Generierung von Prüffällen zu übergeben.

3.6.11.3.2 Anforderungen an die Methodenunterstützung

LSE11.M.1 DNAV-/LOGM-Unterstützung

LSE11.M.1.1 Logische Schemaspezifikationen
Es ist möglich, mit Hilfe der Reihenfolge, Häufigkeit, Priorität und Art der Datenzugriffe auf konzeptionelle Datenmodelle daraus automatisch logische Schemaspezifikationen für ein (relationales, hierarchisches bzw. netzwerkartiges) Teil-Datenbanksystem zu generieren.

LSE11.M.1.2 Sekundärindizes
Es ist möglich, mit Hilfe der Reihenfolge, Häufigkeit, Priorität und Art der Datenzugriffe auf konzeptionelle Datenmodelle daraus automatisch Vorschläge für die Festlegung von Sekundärindizes für ein (relationales, hierarchisches bzw. netzwerkartiges) Teil-Datenbanksystem zu generieren.

LSE11.M.1.3 Vollständigkeit der logischen Schemaspezifikationen
Es ist sichergestellt, daß alle Entitätstypen, Beziehungstypen und Attribute der konzeptionellen Schemaspezifikationen in die logischen Schemaspezifikationen transformiert werden.

LSE11.M.1.4 Übernahme von Integritätsbedingungen
Es ist sichergestellt, daß bei der Erzeugung logischer Schemaspezifikationen die in den konzeptionellen Schemaspezifikationen enthaltenen Integritätsbedingungen in die logischen Schemaspezifikationen übernommen werden.

LSE11.M.1.5 Mehrere logische Schemaspezifikationen
Es ist möglich, logische Schemaspezifikationen für Views (konzeptionelle Teil-Datenmodelle) eines konzeptionellen Datenmodells zu erstellen.

3.6.11.3.3 Anforderungen an die Funktionen

LSE11.F.1 Betriebsmittel- und Zeitbedarf
Es ist möglich, alle Parameter zur Reihenfolge, Häufigkeit und Art der Zugriffe auf Daten – bezüglich verschiedener Verarbeitungseinheiten – zu erfassen. Ebenso können alle weiteren Informationen zum Mengengerüst und der physischen Datenverteilung sowie alle Angaben zum (geforderten) Zeitverhalten erfaßt werden.

3.6.11.3.4 Sonstige Anforderungen

LSE11.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE11.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE11.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.12 Leistungseinheit LSE12 „Komponenten und Module generieren“

3.6.12.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 6.1-SW SW-Module codieren
(„Implementierungsdokumente (SW-Modul).*Code*“,
„Implementierungsdokumente (SW-Modul).*Listen/Protokolle*“ und
„Implementierungsdokumente (SW-Modul).*Prozeduren*“)

Methode

keine

3.6.12.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen aus den Spezifikationen zu Komponenten oder Modulen abhängig von der jeweiligen Zielsprache der Quellcode generiert und abhängig vom jeweiligen Zielsystem der notwendige Prozedurcode erzeugt wird.

3.6.12.3 Anforderungen

3.6.12.3.1 Anforderungen an die Schnittstellen

LSE12.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE12.S.2 Eingangs-Schnittstelle zu LSE10 „Spezifikation der Komponenten und Module unterstützen“
Es ist möglich, mit LSE10 erstellte Spezifikationen für Komponenten und Module ohne weitere Transformation gezielt aus der Objektverwaltung zur Generierung von Quellcode zu übernehmen.

LSE12.S.3 Ausgangs-Schnittstelle zu LSE15 „Kompilieren“
Es ist möglich, generierten Quellcode für Komponenten und Module ohne weitere Transformation über die Objektverwaltung an LSE15 zur Kompilierung zu übergeben.

LSE12.S.4 Ausgangs-Schnittstelle zu LSE17 „Debuggen“
Es ist möglich, Quellcode für Komponenten und Module ohne weitere Transformation über die Objektverwaltung an LSE17 zur Fehlersuche zu übergeben.

LSE12.S.5 Ausgangs-Schnittstelle zu LSE20 „Code rückwärts transformieren“
Es ist möglich, erzeugten Quellcode für Komponenten und Module ohne weitere Transformation über die Objektverwaltung an LSE20 zur Umsetzung in Spezifikationen zu übergeben.

LSE12.S.6 Ausgangs-Schnittstelle zu LQS14 „Prüffälle generieren“
Es ist möglich, generierten Quellcode ohne weitere Transformation über die Objektverwaltung an LQS14 zur Generierung von Prüffällen zu übergeben.

3.6.12.3.2 Anforderungen an die Methodenunterstützung

keine

3.6.12.3.3 Anforderungen an die Funktionen

LSE12.F.1 Generieren von Code
Es ist möglich, Spezifikationen von Komponenten und Modulen abhängig vom jeweiligen Zielsystem in kompilierbaren oder interpretierbaren Code zu transformieren.
Zielsysteme sind hier Compiler oder Interpreter für Quellcode und Prozedurcode bestimmter Hersteller.

LSE12.F.2 Vollständigkeit
Der Code wird so vollständig erzeugt, daß keine manuellen Ergänzungen in nachfolgenden Schritten notwendig sind.

LSE12.F.3 Implizites Verhindern manueller Änderungen
Manuelle Änderungen am generierten Code werden vom Werkzeug implizit verhindert.

LSE12.F.4 Wartung auf Codeebene im Ausnahmefall

LSE12.F.4.1 Explizite Freigabe für manuelle Änderungen
Manuelle Änderungen am generierten Code sind nur nach expliziter Freigabe möglich.
Damit soll eine Wartung auf Codeebene überhaupt erst ermöglicht werden.

LSE12.F.4.2 Modularität
Der generierte Code ist so modular gegliedert, daß Codeabschnitte für die Kommunikation mit dem Nutzer via Nutzeroberfläche, für die Ablaufsteuerung, für die fachliche Verarbeitung und für Daten- bzw. Datenbankzugriffe als in sich abgeschlossene Einheiten identifizierbar sind.

LSE12.F.4.3 Strukturiertheit
Der generierte Code entspricht den Prinzipien der Strukturierten Programmierung.

LSE12.F.5 Einbinden bestehender SW-Komponenten und SW-Module

LSE12.F.5.1 Explizites Einbinden
Es ist möglich, vorhandenen Code für SW-Komponenten und SW-Module während der Generierung explizit in den resultierenden Code einbinden zu lassen.
Im vorliegenden Fall werden nur für den jeweils aktuellen Generierungslauf vom Anwender explizit identifizierte Codeabschnitte in den Code integriert.

LSE12.F.5.2 Automatisches Einbinden
Es ist möglich, existierenden Code für SW-Komponenten und SW-Module während der Generierung automatisch in den resultierenden Code einbinden zu lassen.
Im vorliegenden Fall werden bei allen Generierungsläufen automatisch bestehende Codeabschnitte – z. B. für Anmelde- und Sicherheitsroutinen, Abbruch- und Fehlerbehandlungsroutinen oder Hilferoutinen – in den Code integriert.

LSE12.F.6 Kommentare

LSE12.F.6.1 Kommentare unterdrücken
Es ist möglich, Kommentare (z. B. aus den Spezifikationen) im generierten Code zu unterdrücken.

LSE12.F.6.2 Gliedernde Kommentare
Es ist möglich, gliedernde Kommentare automatisch zu generieren.

LSE12.F.6.3 Ausführlichkeit von Kommentaren
Es ist möglich, das Vorkommen von automatisch generierten Kommentaren in ihrer Häufigkeit und Ausführlichkeit unterschiedlich einzustellen.

3.6.12.3.4 Sonstige Anforderungen

LSE12.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE12.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE12.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.13 Leistungseinheit LSE13 „Datenbanken generieren“

3.6.13.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 6.2-SW Datenbank realisieren
(„Implementierungsdokumente (Datenbank).*Code*“,
„Implementierungsdokumente (Datenbank).*Listen/Protokolle*“ und
„Implementierungsdokumente (Datenbank).*Prozeduren*“)

Methode

keine

3.6.13.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen aus logischen Schemaspezifikationen zu Datenbanken performanceorientierte physische Schemaspezifikationen generiert werden können. Hierbei kann auch der zum Datenbankaufbau notwendige Prozedurcode in der entsprechenden Datendefinitionssprache erzeugt werden.

3.6.13.3 Anforderungen

3.6.13.3.1 Anforderungen an die Schnittstellen

LSE13.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE13.S.2 Eingangs-Schnittstelle zu LSE11 „Spezifikation der Datenbanken unterstützen“
Es ist möglich, mit LSE11 erstellte logische Schemaspezifikationen für Datenbanken und Datenstrukturen ohne weitere Transformation gezielt aus der Objektverwaltung zur Generierung von physischen Schemaspezifikationen zu übernehmen.

LSE13.S.3 Ausgangs-Schnittstelle zu LSE15 „Kompilieren“
Es ist möglich, physische Schemaspezifikationen für Datenbanken und Datenstrukturen ohne weitere Transformation über die Objektverwaltung an LSE15 zur Kompilierung zu übergeben.

LSE13.S.4 Ausgangs-Schnittstelle zu LSE21 „Datenbanken, Datenstrukturen rückwärts transformieren“
Es ist möglich, physische Schemaspezifikationen für Datenbanken und Datenstrukturen ohne weitere Transformation über die Objektverwaltung an LSE21 zur Umsetzung in logische Schemaspezifikationen zu übergeben.

3.6.13.3.2 Anforderungen an die Methodenunterstützung

keine

3.6.13.3.3 Anforderungen an die Funktionen

LSE13.F.1 Erzeugung physischer Schemaspezifikationen
Es ist möglich, mit Hilfe von Eingabeparametern aus den logischen Schemaspezifikationen physische Schemaspezifikationen zu erzeugen (mit Empfehlungen zu Speicherplatzverteilung, Indizes, Puffergrößen, Blockungsfaktoren usw.). Zu den Eingabeparametern gehören Angaben zum Mengengerüst, zum geforderten Zeitverhalten, bezogen auf bestimmte Verarbeitungseinheiten, zur geforderten Datenhaltung bzw. -verteilung und zur Länge und Anzahl der zu verarbeitenden Sätze.

LSE13.F.2 Vollständigkeit der physischen Schemaspezifikationen
Es ist sichergestellt, daß alle Elemente der relevanten logischen Schemaspezifikationen in die physischen Schemaspezifikationen transformiert werden.

LSE13.F.3 Übernahme von Integritätsbedingungen
Es ist sichergestellt, daß bei der Erzeugung physischer Schemaspezifikationen die in den logischen Schemaspezifikationen enthaltenen Integritätsbedingungen in die physischen Schemaspezifikationen übernommen werden.

Die Anforderung bedeutet insbesondere, daß die evtl. notwendige Syntax für entsprechende Integritätsbedingungen in der physischen Schemaspezifikation mit erzeugt wird.

LSE13.F.4 Generierung von Prozedurcode
Es ist möglich, aus den erfaßten logischen Schemaspezifikationen den zum Datenbankaufbau der Zieldatenbank notwendigen Prozedurcode zu generieren.

LSE13.F.5 Versionstreue
Die erzeugten physischen Schemaspezifikationen müssen die Leistungsfaktoren der aktuellen Version der Datenbank berücksichtigen.

Neuere Versionen kommerzieller Datenbanksysteme bringen meist Verbesserungen der Leistungen und Ausdrucksfähigkeiten dieser Systeme mit sich, wie z. B. die Möglichkeit differenziertere Integritätsbedingungen zu formulieren. Dies erfordert meist eine Anpassung der entsprechenden Werkzeuge, die physische Schemaspezifikationen erzeugen.

LSE13.F.6 Berücksichtigung von Performancefaktoren
Die erzeugten physischen Schemaspezifikationen müssen alle Performancefaktoren berücksichtigen, zu denen ausreichende Nutzervorgaben in den logischen Schemaspezifikationen bzw. deren Beschreibungen enthalten sind.

Die Anforderung bedeutet insbesondere, daß die evtl. notwendige Syntax für die physischen Schemaspezifikationen mit erzeugt wird, einschließlich der Datenbankbeschreibungsbefehle zur Erzeugung von Indizes und Vorschläge zum Aufbau des benötigten Speicherplatzes und dessen Verteilung.

3.6.13.3.4 Sonstige Anforderungen

LSE13.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE13.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE13.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.14 Leistungseinheit LSE14 „Nutzerschnittstellenformate generieren“

3.6.14.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 6.1-SW SW-Module codieren
(„Implementierungsdokumente (SW-Modul).*Code*“,
„Implementierungsdokumente (SW-Modul).*Listen/Protokolle*“ und
„Implementierungsdokumente (SW-Modul).*Prozeduren*“)

Methode

keine

3.6.14.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen aus den Spezifikationen von Nutzerschnittstellen (Masken, Listen) die Deklarationen in der jeweiligen Zielsprache generiert werden können.

3.6.14.3 Anforderungen

3.6.14.3.1 Anforderungen an die Schnittstellen

LSE14.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

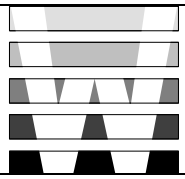
LSE14.S.2 Eingangs-Schnittstelle zu LSE02 „Spezifikation von Nutzerschnittstellen unterstützen“
Es ist möglich, mit LSE02 erstellte Spezifikationen für Schnittstellen ohne weitere Transformation gezielt aus der Objektverwaltung zur Generierung von physischen Schemaspezifikationen zu übernehmen.

LSE14.S.3 Ausgangs-Schnittstelle zu LSE15 „Kompilieren“
Es ist möglich, physische Schemaspezifikationen für Schnittstellen ohne weitere Transformation über die Objektverwaltung an LSE15 zur Kompilierung zu übergeben.

LSE14.S.4 Ausgangs-Schnittstelle zu LSE20 „Code rückwärts transformieren“
Es ist möglich, erzeugte physische Schemaspezifikationen für Schnittstellen über die Objektverwaltung an LSE20 zur Umsetzung in Spezifikationen zu übergeben.

3.6.14.3.2 Anforderungen an die Methodenunterstützung

keine



3.6.14.3.3 Anforderungen an die Funktionen

LSE14.F.1 Maskenstrukturen
Es ist möglich, aus Spezifikationen von Nutzerschnittstellen für Masken in einem TP-Zielsystem die zugehörigen Maskenstrukturen zu generieren.

LSE14.F.2 Listenstrukturen
Es ist möglich, aus Spezifikationen von Nutzerschnittstellen für Listen in einem Zielsystem die zugehörigen Listenstrukturen zu generieren.

LSE14.F.3 Generierungsoptionen

LSE14.F.3.1 Auswahl

LSE14.F.3.1.1 Analyse
Es ist möglich, die Generierungsangaben auf mögliche Fehlersituation hin zu analysieren, ohne die Generierung durchzuführen.

LSE14.F.3.1.2 Unbedingte Generierung
Es ist möglich, die Generierung durchzuführen, ohne mögliche Fehlersituationen zu berücksichtigen.

LSE14.F.3.2 Anzeige
Es ist möglich, Generierungsoptionen anzuzeigen.

LSE14.F.3.3 Voreinstellungen
Es ist möglich, voreingestellte Werte bei der Generierung ohne manuelle Eingriffe zu übernehmen.

LSE14.F.4 Ergebnis
Das Ergebnis der Generierung ist vollständig.

Dies bedeutet, daß das Generierungsergebnis ohne manuelle Ergänzung aus dem Zielsystem verwendbar ist.

3.6.14.3.4 Sonstige Anforderungen

LSE14.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE14.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE14.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.15 Leistungseinheit LSE15 „Kompilieren“

3.6.15.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 6.1-SW SW-Module codieren („SW-Modul“)

– SE 6.2-SW Datenbank realisieren („Datenbank“)

Methode

keine

3.6.15.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen Quellcode in einen Zielcode übersetzt werden kann.

Bei der Übersetzung wird gewährleistet, daß sich die Semantik des Programms nicht ändert. Der Zielcode kann nach schneller Laufzeit oder geringem Speicherbedarf optimiert werden.

3.6.15.3 Anforderungen

3.6.15.3.1 Anforderungen an die Schnittstellen

LSE15.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE15.S.2 Eingangs-Schnittstelle zu LSE12 „Komponenten und Module generieren“
Es ist möglich, mit LSE12 generierten Quellcode für Komponenten und Module ohne weitere Transformation gezielt aus der Objektverwaltung zur Kompilierung zu übernehmen.

LSE15.S.3 Eingangs-Schnittstelle zu LSE13 „Datenbanken generieren“
Es ist möglich, mit LSE13 generierte physische Schemaspezifikationen für Datenbanken ohne weitere Transformation gezielt aus der Objektverwaltung zur Kompilierung zu übernehmen.

LSE15.S.4 Eingangs-Schnittstelle zu LSE14 „Nutzerschnittstellenformate generieren“
Es ist möglich, mit LSE14 generierte physische Schemaspezifikationen für Schnittstellen ohne weitere Transformation aus der Objektverwaltung zur Kompilierung zu übernehmen.

LSE15.S.5 Ausgangs-Schnittstelle zu LSE16 „Binden“
Es ist möglich, kompilierten Code ohne weitere Transformation über die Objektverwaltung an LSE16 zum Binden zu übergeben.

3.6.15.3.2 Anforderungen an die Methodenunterstützung

keine

3.6.15.3.3 Anforderungen an die Funktionen

LSE15.F.1 Fehlerfreie Übersetzung
Ein syntaktisch und semantisch fehlerfreies Programm, das im Quellcode vorliegt, wird in ein semantisch äquivalentes Programm im Zielcode übersetzt.

LSE15.F.2 Ortsangaben

LSE15.F.2.1 Quellcode
Es ist möglich, den Ort des zu übersetzenden Quellcodes anzugeben.

Im allgemeinen steht der Quellcode in einer Datei.

LSE15.F.2.2 Protokoll
Es ist möglich, den Ort des Übersetzungsprotokolls (Listing) anzugeben.

Im allgemeinen wird das Listing in eine Datei geschrieben; es kann aber auch am Bildschirm angezeigt oder in Listenform ausgedruckt werden.

LSE15.F.2.3 Zielcode
Es ist möglich, den Dateinamen mit dem zu erzeugenden Zielcode anzugeben.

LSE15.F.3 Endekennung
Es ist möglich, mittels eines speziellen Zeichens oder einer speziellen Zeichenfolge das Lesen von der Quelle abzubrechen.

In einem Ada-Programm wäre z. B. die Zeichenfolge „—$“ in der ersten Spalte einer Zeile eine mögliche Kennung. Auf diese Weise kann das Ende des Quellcodes auch beim Lesen vom Terminal signalisiert werden.

LSE15.F.4 Übersetzungsprotokoll

LSE15.F.4.1 Varianten
Es ist möglich, für das Layout des Übersetzungsprotokolls zwei Varianten anzugeben: Terminal oder Drucker.

Das Layout für Terminal berücksichtigt die geringere Zeilenbreite (im allgemeinen 80 Zeichen pro Zeile). Das Layout für Drucker hingegen kann die ganze Breite des Druckers (im allgemeinen bis zu 132 Zeichen) ausnutzen.

LSE15.F.4.2 Zusammenfassung
Es ist möglich, eine Zusammenfassung des Übersetzungslaufes am Terminal anzeigen zu lassen.

Die Zusammenfassung enthält Anzahl und Art der Übersetzungsfehler.

LSE15.F.4.3 Fehler
Es ist möglich, im Übersetzungsprotokoll fehlerfreie Zeilen zu unterdrücken.

Es werden nur die fehlerhaften Zeilen mit ihren zugehörigen Fehlermeldungen ausgegeben.

LSE15.F.4.4 Protokollangaben
Das Übersetzungsprotokoll enthält die für die aktuelle Übersetzung relevanten Parameter, die gültigen Optionen und Angaben über Datum und Uhrzeit der Übersetzung.

LSE15.F.4.5 Statistik

LSE15.F.4.5.1 Statistik am Ende
Die Statistik steht am Ende des Übersetzungsprotokolls.

LSE15.F.4.5.2 Quellcode-Zeilen
Die Statistik des Übersetzungsprotokolls enthält die Anzahl der Quellcode-Zeilen.

LSE15.F.4.5.3 Zielcode-Größe
Die Statistik des Übersetzungsprotokolls enthält die Größe des Zielcodes.

Die Größe wird üblicherweise in der kleinsten adressierbaren Maschineneinheit angegeben, also in Byte.

LSE15.F.4.5.4 Zeiten
Die Statistik des Übersetzungsprotokolls enthält die verbrauchte Rechenzeit („elapsed time“) und die reine CPU-Zeit.

LSE15.F.5 Keine Codeerzeugung
Es ist möglich, während der Übersetzung die Codeerzeugung zu unterdrücken.

Es werden nur eventuelle Fehler gemeldet.

LSE15.F.6 Debug-Informationen
Es ist möglich, den Zielcode mit Debug-Informationen zu instrumentieren.

Ein solchermaßen instrumentierter Zielcode kann – falls auch mit der Debug-Option gebunden wird – mit einem Debugger getestet werden.

LSE15.F.7 Test-Modus
Es ist möglich, das Übersetzungsprotokoll mit Informationen zu Testzwecken anzureichern.

Solche Informationen können z. B. sein: Zeitmessungen für die Übersetzung von Anweisungen, erzeugter (Zwischen-) Code in verschiedenen Darstellungen.

LSE15.F.8 Fehlermeldungen

LSE15.F.8.1 Klassifikation
Es existieren mindestens die beiden Klassen „Fehler“ und „Warnung“.

Eine weitergehende Klassifikation würde unterscheiden zwischen „Syntaxfehler“, „semantischer Fehler“, „Warnung“ und „Compiler-Fehler“.

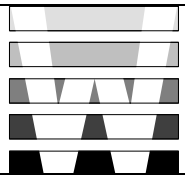
LSE15.F.8.2 Kennzeichnung
Die Fehlerstelle wird im Übersetzungsprotokoll gekennzeichnet.

Dies geschieht z. B. durch einen Zeiger direkt unterhalb der fehlerhaften Stelle im Quellcode.

LSE15.F.8.3 Einstreuen
Die Fehlermeldungen werden im Übersetzungsprotokoll in den Quellcode eingestreut.

LSE15.F.8.4 Maßnahmen
Die Maßnahmen zur Fehlerbehebung werden im Übersetzungsprotokoll aufgeführt.

Beispiel: Beim Auftreten eines Syntaxfehlers werden die automatisch eingefügten Tokens, die zu einem syntaktisch korrekten Programm führen, auf dem Listing aufgeführt.



LSE15.F.8.5 Zielgruppen-orientiert
Es ist möglich, vor der Übersetzung anzugeben, ob die Fehlermeldungen ausführlicher oder knapper sein sollen.

Die ausführlichere Form ist für den Erstnutzer sinnvoll, während die knappere Form für den Sprachexperten gedacht ist, der auf langatmige Erklärungen in den Fehlermeldungen verzichten kann.

LSE15.F.8.6 Fehlerstatistik am Ende
Die Fehlerstatistik steht am Ende des Übersetzungsprotokolls.

LSE15.F.9 Schnittstellen zu Programmiersprachen

LSE15.F.9.1 Assembler
Eine Schnittstelle zur Assembler-Sprache ist vorhanden.

LSE15.F.9.2 Ada
Eine Schnittstelle zu Ada ist vorhanden.

LSE15.F.9.3 FORTRAN
Eine Schnittstelle zu FORTRAN ist vorhanden.

LSE15.F.9.4 COBOL
Eine Schnittstelle zu COBOL ist vorhanden.

LSE15.F.9.5 C
Eine Schnittstelle zu C ist vorhanden.

LSE15.F.9.6 C++
Eine Schnittstelle zu C++ ist vorhanden.

LSE15.F.9.7 Standardkonformität
Die unterstützte Programmiersprache wird standardkonform angewendet.

LSE15.F.9.8 Implementierungsabhängigkeit
Die implementierungsabhängigen Optionen der Programmiersprache sind dokumentiert.

LSE15.F.10 Optimierungen

LSE15.F.10.1 Laufzeit
Es ist möglich, mittels einer Option anzugeben, daß der Code mit dem Ziel möglichst schneller Laufzeit optimiert wird.

Der Speicherbedarf des Zielcodes kann sich dann unter Umständen erhöhen.

LSE15.F.10.2 Speicherbedarf
Es ist möglich, mittels einer Option anzugeben, daß der Code mit dem Ziel möglichst geringen Speicherbedarfs optimiert wird.

Die Laufzeit des Zielcodes kann sich dann unter Umständen erhöhen.

3.6.15.3.4 Sonstige Anforderungen

LSE15.So.1 Fehlermeldungen

LSE15.So.1.1 Verständlichkeit
Die Fehlermeldungen sind klar verständlich und nachvollziehbar.

Falls der Fehler im Übersetzungsprotokoll nicht ausreichend erklärt werden kann, ist auf die entsprechende Stelle in der offiziellen Sprachbeschreibung zu verweisen.

LSE15.So.1.2 Sprache
Die Fehlermeldungen sind sprachlich umgebungskonform verfaßt.

LSE15.So.2 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE15.So.3 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE15.So.4 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LSE15.So.5 Dokumentation
Alle gewählten Implementierungsoptionen des Werkzeugs sind dokumentiert.

3.6.16 Leistungseinheit LSE16 „Binden“

3.6.16.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- SE 6.1-SW SW-Module codieren („SW-Modul“)
- SE 7.1-SW Zur SW-Komponente integrieren („SW-Komponente“)
- SE 7.3-SW Zur SW-Einheit integrieren („SW-Einheit“)
- SE 8.1 Zum System integrieren („System“)

Methode

keine

3.6.16.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit der mehrere Programmeinheiten zu einem ablauffähigen Lademodul zusammengebunden werden können. Dabei werden u. a.

- externe Referenzen aufgelöst und
- die Gültigkeit der einzelnen Einheiten überprüft.

3.6.16.3 Anforderungen

3.6.16.3.1 Anforderungen an die Schnittstellen

LSE16.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE16.S.2 Eingangs-Schnittstelle zu LSE15 „Kompilieren“
Es ist möglich, mit LSE15 kompilierten Code ohne weitere Transformation aus der Objektverwaltung zum Binden zu übernehmen.

LSE16.S.3 Ausgangs-Schnittstelle zu LSE17 „Debuggen“
Es ist möglich, gebundenen Code ohne weitere Transformation über die Objektverwaltung an LSE17 zur Fehlersuche zu übergeben.

LSE16.S.4 Ausgangs-Schnittstelle zu LSE18 „Leistungen abschätzen“
Es ist möglich, gebundenen Code ohne weitere Transformation über die Objektverwaltung an LSE18 zur Abschätzung des voraussichtlichen Leistungsverhaltens zu übergeben.

3.6.16.3.2 Anforderungen an die Methodenunterstützung

keine

3.6.16.3.3 Anforderungen an die Funktionen

LSE16.F.1 Binden

Es ist möglich, mehrere Einheiten zu einem ladbaren und ablauffähigen Programm (Lademodul) zusammenzubinden. Dabei werden die externen Referenzen der Einheiten untereinander aufgelöst und deren Gültigkeit überprüft.

Im allgemeinen werden zu dem Zielcode folgende Einheiten dazugebunden: Speicherverwaltung, Ein-/Ausgabe-Routinen, mathematische Operationen, Ausnahmebehandlung (falls von der Sprache unterstützt), Ablaufsteuerung zur Verwaltung quasi-parallel ablaufender Prozesse (falls von der Sprache unterstützt [englisch: „scheduling"]). Diese dazugebundenen Einheiten sind Teile des Laufzeitsystems.

LSE16.F.2 Ortsangaben

LSE16.F.2.1 Zielcode

Es ist möglich, den Dateinamen mit dem zu bindenden Zielcode anzugeben.

LSE16.F.2.2 Lademodul

Es ist möglich, den Dateinamen mit dem zu generierenden Lademodul anzugeben.

LSE16.F.3 Bindeprotokoll

LSE16.F.3.1 Option

Es ist möglich, mittels einer Option ein Bindeprotokoll erstellen zu lassen.

LSE16.F.3.2 Bindetabelle

Dieses Protokoll enthält eine Bindetabelle („link map").

Die Bindetabelle enthält zumindest eine Liste aller zusammengebundener Einheiten mit ihren Speicheradressen, Größen und Overlaystrukturen (falls vorhanden).

LSE16.F.3.3 Protokollangaben

Das Bindeprotokoll enthält die für den aktuellen Bindevorgang relevanten Parameter, die gültigen Optionen und Angaben über Datum und Urhzeit des Bindens.

LSE16.F.4 Debug-Informationen

Es ist möglich, das Lademodul mit Debug-Informationen zu instrumentieren.

Ein solchermaßen instrumentiertes Lademodul kann – falls es bereits mit der Debug-Option übersetzt wurde – mit einem Debugger getestet werden (siehe LSE17 „Debuggen").

LSE16.F.5 Haldengröße

Es ist möglich, mittels einer Option die Größe der Halde zu spezifizieren.

Mit „Halde" (auch „heap" genannt) bezeichnet man den zur Laufzeit des Lademoduls verwalteten Speicherbereich für dynamisch erzeugte Objekte.

LSE16.F.6 Optimierungen

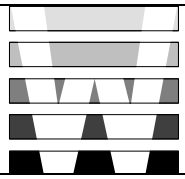
LSE16.F.6.1 Größe

Es ist möglich, mittels einer Option zu veranlassen, daß nicht benutzter Code aus dem Lademodul entfernt wird.

LSE16.F.6.2 Laufzeitroutinen

Nicht benötigte Routinen des Laufzeitsystems werden nicht mit dazugebunden.

Wird zum Beispiel keine Sinus-Funktion verwendet, so braucht sie nicht mit hinzugebunden zu werden und spart somit Speicherplatz.



LSE16.F.6.3 Zeitverhalten
Es ist möglich, mittels einer Option zu veranlassen, daß die zum Lademodul zu bindenden Einheiten derart umgestellt werden, daß das Zeitverhalten des Lademoduls optimiert wird.

Es werden z. B. zusammengehörige Routinen so in Hardwaresegmente verpackt, daß zeitaufwendige Segmentumschaltungen vermieden werden.

LSE16.F.6.4 Verwendungshäufigkeit
Es ist möglich, mittels einer Option anzugeben, daß eine Einheit zur Laufzeit relativ selten verwendet wird.

Eine solche Einheit braucht nicht speicherresident gehalten zu werden.

3.6.16.3.4 Sonstige Anforderungen

LSE16.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE16.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE16.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.17 Leistungseinheit LSE17 „Debuggen“

3.6.17.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 6.3-SW Selbstprüfung des SW-Moduls/der Datenbank durchführen
(„Implementierungsdokumente (SW-Modul)“,
„Implementierungsdokumente (Datenbank)“,
„SW-Modul“ und
„Datenbank“)

Methode

keine

3.6.17.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen aufgetretene Fehler reproduziert und im Quellcode lokalisiert werden können.

Insbesondere sollen dabei

– interne Zustände eines Lademoduls auf Quellcodeebene untersucht werden (z. B. Variablenbelegung, Aufrufstruktur),
– Lademodule auf Quellcode- oder Maschinencodeebene Anweisung für Anweisung durchlaufen werden und
– Lademodule an vorher definierten Halte- und Beobachtungspunkten angehalten werden; Manipulationen sind dann möglich (z. B. an Variablen).

3.6.17.3 Anforderungen

3.6.17.3.1 Anforderungen an die Schnittstellen

LSE17.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE17.S.2 Eingangs-Schnittstelle zu LSE12 „Komponenten und Module generieren“
Es ist möglich, mit LSE12 generierten Quellcode für Komponenten und Module ohne weitere Transformation aus der Objektverwaltung zur Fehlersuche zu übernehmen.

LSE17.S.3 Eingangs-Schnittstelle zu LSE16 „Binden“
Es ist möglich, mit LSE16 gebundenen Code ohne weitere Transformation aus der Objektverwaltung zur Fehlersuche zu übernehmen.

LSE17.S.4 Technische Aufteilung
Falls Entwicklungs- und Zielsystem nicht identisch sind, besteht das Werkzeug aus zwei Teilen; ein Teil läuft auf dem Entwicklungssystem („entwicklungsrechnerabhängiger Teil“ genannt), der andere auf dem Zielsystem („zielrechnerabhängiger Teil“ genannt).

LSE17.S.5 Kommunikation
Entwicklungsrechnerabhängiger und zielrechnerabhängiger Teil kommunizieren miteinander.

Gängige Kopplungsarten sind z. B. eine serielle Verbindung (RS232-Schnittstelle) oder eine Ethernet-Verbindung.

LSE17.S.6 Gleichbleibendes Systemverhalten
Das Systemverhalten wird durch die Veränderung des Lademoduls nicht verändert.

Durch Änderungen im Code des Lademoduls zum Zwecke des Debugging kann sich das Verhalten im ungünstigen Falle so ändern, daß Fehler nicht mehr reproduziert werden können. Außerdem kann sich ein anderes Zeitverhalten ergeben.

3.6.17.3.2 Anforderungen an die Methodenunterstützung

keine

3.6.17.3.3 Anforderungen an die Funktionen

LSE17.F.1 Fehlersuche auf Quellcodeebene

LSE17.F.1.1 Haltepunkte

LSE17.F.1.1.1 Anweisungen
Es ist möglich, Haltepunkte bei den konventionellen Anweisungsarten Zuweisung, Schleifenanweisung, Sprunganweisung und Unterprogrammaufruf zu setzen und zu löschen.

Ein „Haltepunkt“ identifiziert eine Anweisung oder einen Ausdruck im Quellcode. Der Haltepunkt löst aus, wenn die Anweisung oder der Ausdruck ausgeführt wird. Wird ein deaktivierter Haltepunkt durchlaufen, so erfolgt keine Unterbrechung.

LSE17.F.1.1.2 Ausprägungen
Es ist möglich, Haltepunkte innerhalb generischer Ausprägungen zu setzen und zu löschen.

LSE17.F.1.1.3 Inline-Prozeduren
Es ist möglich, Haltepunkte innerhalb von Inline-Prozeduren zu setzen und zu löschen.

LSE17.F.1.1.4 Bedingter Eingangsaufruf
Es ist möglich, Haltepunkte bei bedingten Eingangsaufrufen zu setzen und zu löschen.

LSE17.F.1.1.5 Annahme-Anweisungen
Es ist möglich, Haltepunkte in Annahme-Anweisungen zu setzen und zu löschen.

LSE17.F.1.1.6 Ausnahmebehandler
Es ist möglich, Haltepunkte in Ausnahmebehandlern zu setzen und zu löschen.

LSE17.F.1.1.7 Zeitverzögert
Gesetzte Haltepunkte werden nach einer wählbaren Zeit wirksam, d. h. automatisch aktiviert.

Jeder zusätzliche Haltepunkt erfordert weitere Prozessorleistung zur Überwachung dieses Haltepunktes.

LSE17.F.1.1.8 Durchlaufabhängige Aktivierung
Gesetzte Haltepunkte werden wirksam, wenn eine Anweisung eine wählbare Anzahl von Durchläufen aufweist.

LSE17.F.1.2 Beobachtungspunkte
Es ist möglich, Beobachtungspunkte zu setzen und zu löschen.

Ein „Beobachtungspunkt“ identifiziert einen Operanden. Der Beobachtungspunkt löst aus, wenn die Variable angesprochen wird. Wird ein deaktivierter Beobachtungspunkt durchlaufen, so erfolgt keine Unterbrechung.

LSE17.F.1.3 Einzelschritt-Ausführung

LSE17.F.1.3.1 Aufruf
Es ist möglich, Einzelschritte auf der Ebene Unterprogrammaufruf auszuführen.

LSE17.F.1.3.2 Unterprogrammeintritt/-austritt
Es ist möglich, Einzelschritte auf der Ebene Unterprogrammeintritt und -austritt auszuführen.

LSE17.F.1.3.3 Parameterübergabe
Es ist möglich, Einzelschritte auf der Ebene Parameterübergabe auszuführen.

LSE17.F.1.3.4 Verzweigung/Zuweisung
Es ist möglich, Einzelschritte auf der Ebene Verzweigung und Zuweisung auszuführen.

LSE17.F.1.3.5 Wählbare Anzahl
Es ist möglich, eine wählbare Anzahl von Zeilen und Anweisungen auszuführen.

LSE17.F.1.3.6 Anzeige
Während der Einzelschritt-Ausführung werden die ausgeführten Anweisungen angezeigt.

LSE17.F.1.4 Unterstützung von Spracheigenschaften

LSE17.F.1.4.1 Manipulation symbolischer Variablen
Es ist möglich, symbolische Variablen des Quellprogramms zu lesen und zu schreiben.

LSE17.F.1.4.2 Auswertungen
Ausdrücke der Quellsprache werden ausgewertet.

LSE17.F.1.4.3 Manipulation von Dateien
Es ist möglich, den Inhalt geöffneter Dateien von der Quellcodeebene aus zu ändern.

LSE17.F.1.5 Aufzeichnung

LSE17.F.1.5.1 Ein/aus
Es ist möglich, die Protokollierung einer Debugging-Sitzung ein- und auszuschalten.

Eingaben an das Werkzeug und Ausgaben des Werkzeugs werden aufgezeichnet, damit der Verlauf der Debugging-Sitzung später rekonstruiert werden kann.

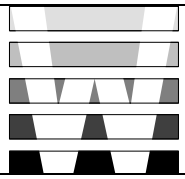
LSE17.F.1.5.2 Programmausgabe
Es ist möglich, die Ausgaben des Lademoduls selbst zu protokollieren.

LSE17.F.1.6 Zustandsausgabe
Es ist möglich, den gegenwärtigen Werkzeugzustand vollständig auszugeben.

Dazu gehören z. B. das Auflisten aller Halte- und Beobachtungspunkte, die Anzeige der Aufrufhierarchie der Unterprogramme, die gegenwärtige Position im Programm, Zustandsinformationen über Prozesse (falls vorhanden) und Angaben über die Speicherbelegung.

LSE17.F.2 Fehlersuche auf Maschinencode-Ebene

LSE17.F.2.1 Maschinencode-Debugger
Die Basisfunktionalität eines Maschinencode-Debuggers ist verfügbar.



Die Basisfunktionalität umfaßt Haltepunkte, Beobachtungspunkte sowie Programmausführung und -kontrolle auf Instruktionsebene, ferner Anzeige des Programmablaufs.

LSE17.F.2.2 Anzeigen und Ausgaben

LSE17.F.2.2.1 Haltepunkte/Beobachtungspunkte
Es ist möglich, Haltepunkte und Beobachtungspunkte bei der Ausgabe von Instruktionen mit anzuzeigen.

LSE17.F.2.2.2 Speicherinhalte

LSE17.F.2.2.2.1 Ausgabe
Die Ausgabe von Speicher- und Registerinhalten ist möglich.

LSE17.F.2.2.2.2 Veränderung
Die Inhaltsveränderung von Speicherbereichen ist möglich.

LSE17.F.2.3 Manipulation von Objektcode

LSE17.F.2.3.1 Assembler-Code
Beim Maschinencode-Debugging wird Assembler-Code benutzt, um Teile des Objektcodes im Speicher zu ersetzen, zu verändern oder zu erweitern.

Eine solche Manipulation am Objektcode wird häufig „Patch“ genannt.

LSE17.F.2.3.2 Maschinennahe Adressierung
Der Zugriffsmechanismus des Werkzeugs ermöglicht – über symbolische Variablen hinaus – die Adressierung jeder Speicherzelle des Programms in der für die jeweilige Maschinenarchitektur kleinstmöglichen Einheit.

Bei Byte-orientierten Maschinenarchitekturen z. B. ist diese Einheit das Byte.

LSE17.F.2.3.3 Sperren manipulierter Objekte
Nach Beendigung einer Debugging-Sitzung werden alle Objekte, deren Instruktionen manipuliert worden sind, für folgende nicht-privilegierte Zugriffe gesperrt.

Damit wird sichergestellt, daß der angezeigte Quelltext mit dem zum Ablauf des Lademoduls verwendeten Quelltext übereinstimmt.

LSE17.F.2.3.4 Meldung
Vor der ersten Manipulation von Objektcode wird eine Meldung ausgegeben, die darauf hinweist, daß dieses Objekt nach Beendigung der Debugging-Sitzung für folgende nicht-privilegierte Zugriffe gesperrt werden wird.

LSE17.F.3 Gestaltung einer Debugging-Sitzung

LSE17.F.3.1 Stop

LSE17.F.3.1.1 Vor Bedingung
Es ist möglich, den Programmablauf unmittelbar vor einer erfüllten Bedingung im Quellprogramm anzuhalten.

LSE17.F.3.1.2 Nach Bedingung
Es ist möglich, den Programmablauf unmittelbar nach einer erfüllten Bedingung im Quellprogramm anzuhalten.

LSE17.F.3.2 Überspringen
Es ist möglich, bei der Programmausführung Programmteile zu überspringen.

Es wird die nächste Quellzeile oder Quellanweisung nach dem übersprungenen Programmteil ausgeführt.

LSE17.F.3.3 Kommandodateien
Debugger-Kommandodateien sind während einer Sitzung ausführbar.

Damit kann eine Anzahl von Kommandos automatisch ausgeführt werden, ohne daß sie bei jedem weiteren Programmablauf neu von Hand eingegeben werden müssen.

LSE17.F.3.4 Automatisch
Es ist möglich, eine aufgezeichnete Debugging-Sitzung (siehe LSE17.F.1.5.1) in Form einer Debugger-Kommandodatei abzuspeichern.

LSE17.F.3.5 Datei-Ein/Ausgabe
Es ist möglich, die Ein/Ausgabe des Lademoduls – sofern sinnvoll – auf andere Dateien umzulenken.

Beim Lesen von einem hardwarenahen Gerät (z. B. Analog-Digital-Wandler) ist eine solche Umlenkung auf eine Datei als Eingabequelle nicht sinnvoll.

LSE17.F.3.6 Arbeitszustand
Es ist möglich, eine Debugging-Sitzung zu unterbrechen und mit dem letzten Arbeitszustand wieder aufzunehmen.

LSE17.F.3.7 Neustart
Es ist möglich, einen Lademodul ohne Neuladen zu starten, falls der Objektcode nicht manipuliert wurde.

Beim wiederholten Laden des Lademoduls wird so Zeit eingespart.

LSE17.F.3.8 Automatisch
Es ist möglich, an einem Haltepunkt oder Beobachtungspunkt den automatischen Ablauf einer Debugger-Kommandofolge mit allen Arten von Debugger-Kommandos auszulösen.

LSE17.F.3.9 Prozeßverhalten
Es ist möglich festzulegen, ob während einer Unterbrechung aufgrund eines Halte- oder Beobachtungspunktes alle Prozesse ruhen oder ob nur das unterbrochene Programmstück ruht und die anderen Prozesse normal weiterlaufen.

Die zweite Variante ist sinnvoll, wenn im Echtzeitbetrieb Eingabedaten innerhalb gewisser Zeitspannen verarbeitet werden müssen.

LSE17.F.4 Normalausführung

Ein für Debugging compiliertes Programm ist auch ohne Debugging ausführbar.

LSE17.F.5 Ada-spezifische Anforderungen

LSE17.F.5.1 Ada-Prozesse

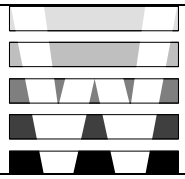
LSE17.F.5.1.1 Prioritäten
Es ist möglich, die Priorität von Ada-Prozessen während ihres Ablaufs abzuändern.

LSE17.F.5.1.2 Starten/Beenden
Es ist möglich, Ada-Prozesse zu starten und zu beenden.

LSE17.F.5.2 Ada-Ausnahmen

LSE17.F.5.2.1 Explizite Auslösung
Es ist möglich, Ada-Ausnahmen explizit auszulösen.

LSE17.F.5.2.2 Ausnahmen ohne Folgen
Es ist möglich, Ada-Ausnahmen ignorieren zu lassen.



LSE17.F.5.3 Ada-Semantik

LSE17.F.5.3.1 Navigation in Prozessen/Unterprogrammen

LSE17.F.5.3.1.1 Prozesse
Es ist möglich, in der Menge aktiver und ruhender Ada-Prozesse zu navigieren.

LSE17.F.5.3.1.2 Unterprogramme
Es ist möglich, in der Menge aufgerufener und noch nicht abgeschlossener Unterprogramme geeignet zu navigieren.

Beispielsweise kann zum statischen oder auch dynamischen Vorgänger eines Unterprogramms navigiert werden.

LSE17.F.5.3.1.3 Pakete
Es ist möglich, in der Menge der umgebenden Pakete zu navigieren.

LSE17.F.5.3.2 Navigation im Namensraum

LSE17.F.5.3.2.1 Identifikation
Es ist möglich, bei der Navigation die zugehörigen Programmeinheiten zu identifizieren.

LSE17.F.5.3.2.2 Lesen und Schreiben
Es ist möglich, bei der Navigation die im Namensraum enthaltenen Variablen zu lesen und zu schreiben.

LSE17.F.5.3.3 Anfragen

LSE17.F.5.3.3.1 Inneres Quellkonstrukt
Es ist möglich, auf Anfrage einen Hinweis auf das laufende innere Quellkonstrukt zu erhalten.

LSE17.F.5.3.3.2 Übersetzungseinheit
Es ist möglich, auf Anfrage einen Hinweis auf die laufende Ada-Übersetzungseinheit zu erhalten.

LSE17.F.5.3.3.3 Langname
Es ist möglich, auf Anfrage einen Hinweis auf den vollen Namen des innersten benannten, aktuellen Ada-Konstrukts zu erhalten.

LSE17.F.5.3.4 Sichtbarkeitsregeln
Die Sichtbarkeitsregeln der Programmiersprache Ada werden berücksichtigt.

LSE17.F.5.3.5 Überladene Variable
Es ist möglich, auf überladene Ada-Variable per Kontextangabe zuzugreifen.

LSE17.F.5.3.6 Elaborationscode
Es ist möglich, die Abarbeitung von Vereinbarungen („Elaboration“ genannt) zu kontrollieren.

Nach der Elaboration einer Vereinbarung hat die Vereinbarung Gültigkeit. Zum Beispiel: Nach der Elaboration einer Variablendeklaration wird die Variable mit dem deklarierten Typ erzeugt, so daß sie nunmehr ansprechbar ist.

LSE17.F.6 Konsistenzprüfung

LSE17.F.6.1 Disassemblierung
Es ist möglich, den ausführbaren Code rückwärts in Assemblersprache zu transformieren.

LSE17.F.6.2 Inkonsistenzen
Es ist möglich, den (assemblierten) Quellcode und den disassemblierten ausführbaren Code auf Inkonsistenzen zu untersuchen.

3.6.17.3.4 Sonstige Anforderungen

LSE17.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE17.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE17.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.18 Leistungseinheit LSE18 „Leistungen abschätzen“

3.6.18.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 2.3 Realisierbarkeit untersuchen
(„Systemarchitektur.*Realisierbarkeitsuntersuchungen*“)

– SE 5.2-SW Betriebsmittel- und Zeitbedarf analysieren
(„SW-Entwurf.*Kenngrößen*“)

Methode

keine

3.6.18.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen eine Analyse des Leistungsverhaltens von Komponenten, Modulen und Datenbanken durchgeführt werden kann.

3.6.18.3 Anforderungen

3.6.18.3.1 Anforderungen an die Schnittstellen

LSE18.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE18.S.2 Eingangs-Schnittstelle zu LSE16 „Binden“
Es ist möglich, mit LSE16 gebundenen Code ohne weitere Transformation gezielt aus der Objektverwaltung zur Abschätzung des voraussichtlichen Leistungsverhaltens zu übernehmen.

3.6.18.3.2 Anforderungen an die Methodenunterstützung

keine

3.6.18.3.3 Anforderungen an die Funktionen

LSE18.F.1 Erfassung von Meßergebnissen

LSE18.F.1.1 Sensoren
Es ist möglich, den zu untersuchenden Teil der Anwendung durch Sensoren zu instrumentieren, die relevante Ereignisse erkennen und anzeigen.

Sensoren erzeugen Meßdatensätze, die von nachfolgenden Komponenten des Meßsystems weiterverarbeitet werden.

LSE18.F.1.2 Meßgrößen
Es ist möglich, Ereignissen Meßgrößen zuzuordnen.

LSE18.F.1.3 Abstraktionsniveau
Es ist möglich, Messungen auf verschiedenen Abstraktionsniveaus durchzuführen.

Verschiedene Abstufungen bzgl. der Granularität der Meßgrößen bewirken einen unterschiedlichen Umfang zu analysierender Informationen und entsprechen damit einem speziellen Abstraktionsniveau.

LSE18.F.1.4 Aktivieren/Deaktivieren
Es ist möglich, Sensoren jederzeit zu aktivieren oder zu deaktivieren.

LSE18.F.1.5 Ereignisbehandlung
Es ist möglich, jedem Sensor eine eigene Ereignisbehandlungsroutine zuzuordnen.

LSE18.F.2 Auswertung von Meßergebnissen

LSE18.F.2.1 Interferenz
Die Auswertung der Meßergebnisse hat keinen Einfluß auf die zu messenden Größen, bzw. dieser Einfluß ist bekannt, und das Ergebnis kann dementsprechend korrigiert werden.

Die Auswertung der Meßergebnisse kann zu Leistungsverlust und zu ungenauen Werten führen. Aus diesem Grund sollte die Auswertung auf eigenen Ressourcen des Meßsystems (z. B. auf eigener Hardware) erfolgen.

LSE18.F.2.2 Kollektoren
Es ist möglich, logisch zusammenhängende Meßdatensätze der Sensoren zusammenzufassen und neue Meßdatensätze für die anschließende Auswertung zu erzeugen.

LSE18.F.3 Darstellung von Meßergebnissen

LSE18.F.3.1 Auswerter
Es ist möglich, Meßdaten in Kenngrößen umzuwandeln und grafisch darzustellen.

Die Umrechnung von Meßdaten in Kenngrößen erlaubt die Anzeige der Ergebnisse in Form von Torten- und Balkendiagrammen oder anderen grafischen Repräsentationsformen.

LSE18.F.3.2 Online
Es ist möglich, sich Kenngrößen online am Bildschirm anzeigen zu lassen.

LSE18.F.3.3 Offline
Es ist möglich, Kenngrößen zu sammeln und die Ergebnisse im nachhinein offline auszuwerten.

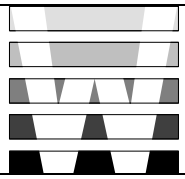
LSE18.F.3.4 Anzeige
Es ist möglich, zwischen verschiedenen Darstellungsformen der Kenngrößen zu wählen.

LSE18.F.4 Lastgenerator
Es steht ein Lastgenerator zur Verfügung, der die natürliche Umgebung der gesamten zu untersuchenden Anwendung oder einzelner Teile ersetzt.

Ein Lastgenerator ist erforderlich, wenn die zu untersuchende Anwendung prototypischen Charakter besitzt und/oder eine Leistungsmessung unter Produktionsbedingungen nicht möglich ist. Ein Lastgenerator stellt kein konkretes Werkzeug dar. Er kann beispielsweise einen Teil einer Anwendung für Meßzwecke ersetzen, um das spätere Verhalten der Anwendung im Produktionsbetrieb zu beobachten.

LSE18.F.5 Zeitbasis
Es steht eine Zeitbasis ausreichender Auflösung zur Verfügung.

Die Zeitbasis wird idealerweise durch eine Hardware-Uhr realisiert.



3.6.18.3.4 Sonstige Anforderungen

LSE18.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE18.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE18.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.19 Leistungseinheit LSE19 „Zeitverhalten simulieren“

3.6.19.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- SE 2.3 Realisierbarkeit untersuchen („Systemarchitektur.*Realisierbarkeitsuntersuchungen*“)
- SE 5.2-SW Betriebsmittel- und Zeitbedarf analysieren („SW-Entwurf.*Kenngrößen*“)

Methode

- Simulationsmodelle (SIMU)

3.6.19.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen Vorgänge auf einer Rechenanlage nachgebildet werden können, die aus Zeit-, Kosten- und anderen Gründen nicht tatsächlich durchgeführt werden können. Hierunter fällt z. B. nicht nur die Simulation einer Flugzeugsteuerung, sondern auch die Simulation von Software im Planungsstadium, wie sie in den

- Spezifikationen von Schnittstellen,
- Prozeßmodellen und
- logischen Schemaspezifikationen von Datenbanken

zum Ausdruck kommt.

Die Simulation von Software setzt voraus, daß konkrete Daten darüber vorliegen,

- wieviele Berechnungen für einen Vorgang durchzuführen sind[1],
- wieviele Zugriffe auf Speicher notwendig sind,
- welche Datenmengen zwischen Hauptspeicher und Prozessor bewegt werden müssen,
- wieviele Zugriffe auf Plattenlaufwerke notwendig sind,
- welche Datenmengen zwischen Festplatte und Hauptspeicher bewegt werden müssen.

Für das simulierte Rechnersystem muß bekannt sein:

- die Rechenleistung des Prozessors (Takte pro Anweisung, Anweisungen pro Sekunde),
- die Übertragungsleistung zwischen Prozessor und Hauptspeicher,
- die Übertragungsleistung zwischen Rechner und Plattenspeicher.

Bei der Simulation von Echtzeit-Systemen kommt die Berücksichtigung der Peripheriegeräte und deren Reaktionszeiten sowie die Eingabe- und Reaktionszeit für Bedienereingaben hinzu.

---

[1] Beispiel: Bei der Simulation von Datenbanken ist der Rechenaufwand sehr stark abhängig von der Art des Vorgangs; die Änderung des aktuellen Datensatzes bedeutet gewöhnlich bedeutend weniger Aufwand als das Suchen eines Datensatzes oder gar das Regenerieren eines Indexes.

3.6.19.3 Anforderungen

3.6.19.3.1 Anforderungen an die Schnittstellen

LSE19.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE19.S.2 Eingangs-Schnittstelle zu LSE02 „Spezifikation von Nutzerschnittstellen unterstützen“
Es ist möglich, mit LSE02 erstellte Spezifikationen für Schnittstellen ohne weitere Transformation gezielt aus der Objektverwaltung zur Simulation des Zeitverhaltens zu übernehmen.

LSE19.S.3 Eingangs-Schnittstelle zu LSE04 „Prozeßmodellierung unterstützen“
Es ist möglich, mit LSE04 erstellte Prozeßmodelle ohne weitere Transformation gezielt aus der Objektverwaltung zur Simulation des Zeitverhaltens zu übernehmen.

LSE19.S.4 Eingangs-Schnittstelle zu LSE11 „Spezifikation der Datenbanken unterstützen“
Es ist möglich, mit LSE11 erstellte logische Schemaspezifikationen für Datenbanken und Datenstrukturen ohne weitere Transformation gezielt aus der Objektverwaltung zur Simulation des Zeitverhaltens zu übernehmen.

3.6.19.3.2 Anforderungen an die Methodenunterstützung

LSE19.M.1 Simulationsmodelle (SIMU)

LSE19.M.1.1 Ressourcen

LSE19.M.1.1.1 Einschränkbar
Es ist möglich, die Verfügbarkeit von Ressourcen einzuschränken.

Ressourcen sind z. B. CPU, Bus, Speicher, Controller, Festplatten, Peripherie; z. B. kann bei einem Multitasking-Betriebssystem ein einzelnes Programm nur über Teile der Gesamtkapazität verfügen.

LSE19.M.1.1.2 Exklusiv verfügbar
Es ist möglich, Ressourcen jeweils nur einem Vorgang zur Verfügung zu stellen.

LSE19.M.1.1.3 Gleichzeitige Benutzung
Es ist möglich, Ressourcen verschiedenen Vorgängen zur gleichzeitigen Benutzung zur Verfügung zu stellen.

LSE19.M.1.1.4 Nutzungsregeln
Es ist möglich, die Nutzung von Ressourcen nach einem bestimmten Verfahren zu regeln.

Beispielsweise gibt es Regeln für den Zugriff auf den Bus eines Rechners.

LSE19.M.1.1.5 Abbildbarkeit
Es ist möglich, Ressourcen uneingeschränkt abzubilden.

LSE19.M.1.1.6 Verfügbarkeitseinschränkungen
Es ist möglich, alle Verfügbarkeitseinschränkungen zu modellieren.

LSE19.M.1.2 Warteschlangen

LSE19.M.1.2.1 Abbildbarkeit
Es ist möglich, Warteschlangen abzubilden.

LSE19.M.1.2.2 Beschränkung
Es ist möglich, die Kapazität von Warteschlangen zu beschränken.

Beispielsweise kann der Befehlsspeicher eines Rechners begrenzt sein.

LSE19.M.1.2.3 Prioritätsregeln
Es ist möglich, Prioritätsregeln für Warteschlangen vorzugeben.

LSE19.M.1.2.4 Prioritätsänderung
Es ist möglich, daß sich die Prioritäten in einer Warteschlange im Zeitablauf oder durch äußere Einflüsse ändern.

LSE19.M.1.2.5 Zustandsabfrage
Der Zustand einer Warteschlange ist jederzeit abfragbar.

LSE19.M.1.3 Aktivitäten

LSE19.M.1.3.1 Feste Zeitvorgabe
Es ist möglich, den Zeitbedarf einer Aktivität fest vorzugeben.

LSE19.M.1.3.2 Berechnung von Zeitvorgaben
Es ist möglich, den Zeitbedarf einer Aktivität aus beliebigen Formeln zu berechnen.

LSE19.M.1.3.3 Aktivitätsabhängiger Zeitbedarf
Es ist möglich, daß der Zeitbedarf einer Aktivität von deren Eigenschaften abhängt.

LSE19.M.1.3.4 Äußere Einflüsse
Es ist möglich, daß der Zeitbedarf einer Aktivität von äußeren Einflüssen bestimmt wird.

LSE19.M.1.3.5 Stochastische Verteilung
Es ist möglich, daß der Zeitbedarf für die Ausführung einer Aktivität stochastisch verteilt ist.

LSE19.M.1.3.6 Beliebige Kombinationen
Es ist möglich, die oben genannten Einflußfaktoren für Aktivitäten beliebig zu kombinieren.

LSE19.M.1.4 Logische Verknüpfungen

LSE19.M.1.4.1 UND/ODER
Es ist möglich, mindestens die logischen Verknüpfungen „UND“ und „ODER“ vorzugeben.

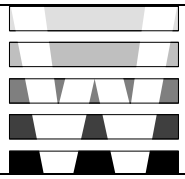
LSE19.M.1.4.2 Einfluß auf Ressourcen
Es ist möglich, daß logische Verknüpfungen die Verfügbarkeit von Ressourcen einschränken.

LSE19.M.1.4.3 Einfluß auf Abarbeitungsreihenfolgen
Es ist möglich, daß logische Verknüpfungen die Abarbeitungsreihenfolgen bestimmen.

LSE19.M.1.5 Ereignisse

LSE19.M.1.5.1 Äußere Eingriffe
Es ist möglich, über Ereignisse äußere Eingriffe zu simulieren.

Äußere Eingriffe sind z. B. Nutzereingaben, Hardware-Unterbrechungen, Software-Unterbrechungen.



LSE19.M.1.5.2 Systemzustandsänderung
Es ist möglich, daß Ereignisse Systemzustände ändern.
Globale, auch kontinuierliche Variable können sich ändern.

LSE19.M.1.5.3 Beeinflussung der Ressourcenverfügbarkeit
Es ist möglich, daß Ereignisse die Verfügbarkeit von Ressourcen beeinflussen.
Der Ausfall von Systemteilen kann so simuliert werden.

LSE19.M.1.5.4 Vorgangsauslösung
Es ist möglich, daß Ereignisse neue Vorgänge auslösen.

LSE19.M.1.5.5 Auslösung durch Vorgang
Es ist möglich, daß Ereignisse durch Vorgänge ausgelöst werden.
Beispielsweise kann ein Vorgang eine Schaltung auslösen.

LSE19.M.1.5.6 Auslösung durch Zustandsänderungen
Es ist möglich, daß Ereignisse durch Zustandsänderungen ausgelöst werden.
Bei Überqueren eines Schwellenwertes einer kontinuierlichen Zustandsgröße kann ein Ereignis ausgelöst werden.

LSE19.M.1.6 Zustandsänderungen

LSE19.M.1.6.1 Kontinuierlich
Es ist möglich, sich kontinuierlich ändernde Systemzustände zu beschreiben.

LSE19.M.1.6.2 Lineare Gleichungen
Es ist möglich, Zustandsänderungen durch lineare Gleichungen zu beschreiben.

LSE19.M.1.6.3 Differentialgleichungen
Es ist möglich, Zustandsänderungen durch Differentialgleichungen zu beschreiben.

LSE19.M.1.6.4 Stellung von Lösungsverfahren
Es werden Lösungsverfahren für Gleichungen zur Verfügung gestellt.

LSE19.M.1.6.5 Beeinflussung durch Ereignisse
Es ist möglich, Systemzustände durch Ereignisse zu beeinflussen.

LSE19.M.1.6.6 Beeinflussung durch globale Variablen
Es ist möglich, Systemzustände durch globale Variablen zu beeinflussen.
Auslastungsgrößen von Ressourcen können sich auf das Systemverhalten auswirken.

LSE19.M.1.6.7 Beeinflussung durch lokale Variablen
Es ist möglich, Systemzustände durch lokale Variablen zu beeinflussen.
Lokale Variablen sind z. B. Attribute von Vorgängen.

LSE19.M.1.6.8 Beeinflussung von Aktivitätsdauern
Es ist möglich, daß Systemzustände Aktivitätsdauern beeinflussen.

LSE19.M.1.6.9 Unabhängige Entwicklung
Es ist möglich, daß Systemzustände sich unabhängig von den modellierten Zusammenhängen entwickeln.
Dies ermöglicht die Beschreibung äußerer Einflüsse, z. B. bei Systemen der Meß- und Regeltechnik.

LSE19.M.1.7 Unterbrechungen

LSE19.M.1.7.1 Jederzeitige Unterbrechbarkeit
Die Simulation ist jederzeit unterbrechbar.

LSE19.M.1.7.2 Manuell
Es ist möglich, Unterbrechungen manuell herbeizuführen.

LSE19.M.1.7.3 Vorbestimmt
Es ist möglich, Unterbrechungen vorzubestimmen.

LSE19.M.1.7.4 Zeitorientiert
Es ist möglich, daß sich Unterbrechungen an der Simulationszeit orientieren.

LSE19.M.1.7.5 Zustandsorientiert
Es ist möglich, daß sich Unterbrechungen an Systemzuständen orientieren.

LSE19.M.1.7.6 Ereignisorientiert
Es ist möglich, daß Unterbrechungen mit Ereignissen gekoppelt werden.
Das Anhalten der Simulation bei Eintritt eines bestimmten Ereignisses ist möglich.

LSE19.M.1.7.7 Zustandsdarstellung
Es ist möglich, während einer Unterbrechung alle Zustände darzustellen.
Beispielsweise können Variablen abgefragt werden.

LSE19.M.1.7.8 Zustandsänderung
Es ist möglich, Systemzustände manuell zu ändern.
Beispielsweise können Variablen oder Parameter geändert werden.

LSE19.M.1.7.9 Ablaufänderung
Es ist möglich, in den Simulationsablauf einzugreifen.
Vorgänge können unterbrochen oder neu ausgelöst werden.

LSE19.M.1.7.10 Schrittweise Fortsetzung
Es ist möglich, die Simulation nach einer Unterbrechung schrittweise fortzusetzen.

LSE19.M.1.7.11 Uneingeschränkte Fortsetzung
Es ist möglich, die Simulation nach einer Unterbrechung uneingeschränkt fortzusetzen.

LSE19.M.1.8 Zustandssicherung

LSE19.M.1.8.1 Abspeichern
Es ist möglich, Simulationszustände abzuspeichern.

LSE19.M.1.8.2 Zurückladen
Es ist möglich, Simulationszustände zurückzuladen.

LSE19.M.1.9 Protokollierung
Es ist möglich, eine Ablauf-Liste aller Vorgänge, Aktivitäten und eingetretenen Ereignisse erzeugen zu lassen.

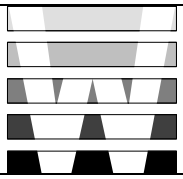
3.6.19.3.3 Anforderungen an die Funktionen

LSE19.F.1 Animation
Es ist möglich, die Simulation durch eine Animation zu visualisieren.

3.6.19.3.4 Sonstige Anforderungen

LSE19.So.1 Makrosprache
Es ist möglich, mit einer Makrosprache den Simulationsablauf zu kontrollieren.

LSE19.So.2 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.



LSE19.So.3 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE19.So.4 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.20 Leistungseinheit LSE20 „Code rückwärts transformieren“

3.6.20.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 4.1-SW SW-Architektur entwerfen
(„SW-Architektur.*Modularisierung*“)

– SE 5.1-SW SW-Komponente/-Modul/Datenbank beschreiben
(„SW-Entwurf.*SW-Komponenten-/SW-Modul-Beschreibung*“)

Methode

keine

3.6.20.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen Nachdokumentation und Analyse des Codes einer Implementierung durchgeführt werden können. Hierbei ergeben sich:

– die statische Zerlegung in SW-Komponenten und SW-Module,
– die Einzelbeschreibungen der SW-Komponenten und SW-Module und
– die Schnittstellen der SW-Komponenten und SW-Module.

3.6.20.3 Anforderungen

3.6.20.3.1 Anforderungen an die Schnittstellen

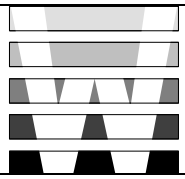
LSE20.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE20.S.2 Eingangs-Schnittstelle zu LSE12 „Komponenten und Module generieren“
Es ist möglich, mit LSE12 erzeugten Quellcode für Komponenten und Module ohne weitere Transformation aus der Objektverwaltung zu übernehmen.

LSE20.S.3 Eingangs-Schnittstelle zu LSE14 „Nutzerschnittstellenformate generieren“
Es ist möglich, mit LSE14 erzeugte physische Schemaspezifikationen für Schnittstellen ohne weitere Transformation aus der Objektverwaltung zu übernehmen.

LSE20.S.4 Ausgangs-Schnittstelle zu LSE02 „Spezifikation von Nutzerschnittstellen unterstützen“
Es ist möglich, Spezifikationen, die aus den physischen Schemaspezifikationen für Schnittstellen erzeugt wurden, ohne weitere Transformation über die Objektverwaltung an LSE02 zu übergeben.

LSE20.S.5 Ausgangs-Schnittstelle zu LSE10 „Spezifikation der Komponenten und Module unterstützen“
Es ist möglich, Spezifikationen, die aus dem Quellcode für Komponenten und Module erzeugt wurden, ohne weitere Transformation über die Objektverwaltung an LSE10 zu übergeben.



3.6.20.3.2 Anforderungen an die Methodenunterstützung

keine

3.6.20.3.3 Anforderungen an die Funktionen

LSE20.F.1 Code-Bearbeitung
Es ist möglich, den Code zurückzuführen, ohne Vorbereitungsarbeiten am Code zusätzlich durchführen zu müssen.

LSE20.F.2 Rückführungsreihenfolge
Es ist möglich, den Informationsgehalt des zu transformierenden Codes vollständig in die neue Struktur zu überführen.

LSE20.F.3 Voreinstellung
Es ist möglich, voreingestellte Werte zur Steuerung der Rückführung ohne manuelle Eingriffe zu übernehmen.

LSE20.F.4 Rückführungsoptionen

LSE20.F.4.1 Auswahl
Es ist möglich, zumindest zwischen den Optionen „Analyse“ und „unbedingte Rückführung“ zu wählen.

LSE20.F.4.2 Anzeige
Es ist möglich, Rückführungsoptionen anzuzeigen.

LSE20.F.5 Ergebnis
Es ist möglich, das Rückführungsergebnis ohne manuelle Ergänzung zu verwenden.

LSE20.F.6 Objekte
Es ist möglich, daß jedes Objekt einer Coderückführung als Objekt angelegt wird.

LSE20.F.7 Platzhalter
Es ist möglich, daß Aufrufe an Programme, die nicht zurückgeführt werden, als leeres Objekt (Platzhalter) erzeugt werden.

LSE20.F.8 Datendeklaration
Es ist möglich, daß Datendeklarationen eines Programmes zurückgeführt werden.

LSE20.F.9 Analyse/Statistiken

LSE20.F.9.1 Call-Analyse
Es ist möglich, daß der Code bezüglich der Call-Aufrufe analysiert und das Ergebnis dokumentiert wird.

LSE20.F.9.2 Transfer-of-Control-Analyse
Es ist möglich, daß der Code bezüglich der Steuerungsübergabeanweisungen (Transfer-of-Control) analysiert und das Ergebnis dokumentiert wird.

LSE20.F.9.3 Verbanalyse
Es ist möglich, daß der Code bezüglich der verwendeten Verben analysiert und das Ergebnis dokumentiert wird.

LSE20.F.9.4 Anweisungsabdeckung
Es ist möglich, nicht durchlaufenen Code zu analysieren und zu dokumentieren.

LSE20.F.10 Grafische Ausgaben

LSE20.F.10.1 Aufrufhierarchiediagramm
Es ist möglich, daß der zurückgeführte Code mittels eines Aufrufhierarchiediagrammes dargestellt wird.

LSE20.F.10.2 Modulverknüpfungsdiagramm
Es ist möglich, daß der zurückgeführte Code mittels eines Modulverknüpfungsdiagrammes dargestellt wird.

LSE20.F.11 Verweisliste
Es ist möglich, daß der zurückgeführte Code mittels einer Verweisliste die Verwendung der Objekte und Verben darstellt.

3.6.20.3.4 Sonstige Anforderungen

LSE20.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE20.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE20.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.21 Leistungseinheit LSE21 „Datenbanken, Datenstrukturen rückwärts transformieren“

3.6.21.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 1.5 System fachlich strukturieren („Anwenderforderungen.*Beschreibung der Funktionalität*“)

– SE 3.3 Anforderungen an die Funktionalität definieren („Technische Anforderungen.*Gesamtfunktion*“)

– SE 5.1-SW SW-Komponente/-Modul/Datenbank beschreiben („SW-Entwurf.*Datenbankbeschreibung*“ und „Datenkatalog“)

Methode

keine

3.6.21.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen aus logischen bzw. physischen Schemaspezifikationen für Datenbanken oder Dateien konzeptionelle Schemaspezifikationen erstellt werden können. Mehrere konzeptionelle Schemaspezifikationen lassen sich zu einer gemeinsamen konsistenten konzeptionellen Schemaspezifikation zusammenfassen.

3.6.21.3 Anforderungen

3.6.21.3.1 Anforderungen an die Schnittstellen

LSE21.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE21.S.2 Eingangs-Schnittstelle zu LSE09 „Informationsstrukturierung unterstützen“
Es ist möglich, mit LSE09 erstellte Informationsstrukturen ohne weitere Transformation aus der Objektverwaltung zur Berücksichtigung bei der Analyse von Beschreibungen für Datenbanken und Datenstrukturen zu übernehmen.

LSE21.S.3 Eingangs-Schnittstelle zu LSE11 „Spezifikation der Datenbanken unterstützen“
Es ist möglich, mit LSE11 erstellte logische Schemaspezifikationen für Datenbanken und Datenstrukturen ohne weitere Transformation aus der Objektverwaltung zur Umsetzung in Informationsstrukturen zu übernehmen.

LSE21.S.4 Eingangs-Schnittstelle zu LSE13 „Datenbanken generieren“
Es ist möglich, mit LSE13 erzeugte physische Schemaspezifikationen für Datenbanken und Datenstrukturen ohne weitere Transformation aus der Objektverwaltung zur Umsetzung in logische Schemaspezifikationen zu übernehmen.

LSE21.S.5 Ausgangs-Schnittstelle zu LSE09 „Informationsstrukturierung unterstützen“
Es ist möglich, Informationsstrukturen, die aus logischen Schemaspezifikationen für Datenbanken und Datenstrukturen erzeugt wurden, ohne weitere Transformation über die Objektverwaltung an LSE09 zu übergeben.

LSE21.S.6 Ausgangs-Schnittstelle zu LSE11 „Spezifikation der Datenbanken unterstützen“
Es ist möglich, logische Schemaspezifikationen für Datenbanken und Datenstrukturen, die aus physischen Schemaspezifikationen erzeugt wurden, ohne weitere Transformation über die Objektverwaltung an LSE11 zu übergeben.

3.6.21.3.2 Anforderungen an die Methodenunterstützung

keine

3.6.21.3.3 Anforderungen an die Funktionen

LSE21.F.1 Transformation logischer in konzeptionelle Schemaspezifikationen
Es ist möglich, eine logische Schemaspezifikation für Datenbanken in eine konzeptionelle Schemaspezifikation zu transformieren.

LSE21.F.2 Vollständigkeit der Transformation
Es ist sichergestellt, daß bei der Transformation einer logischen in eine konzeptionelle Schemaspezifikation alle in der logischen Schemaspezifikation vorhandenen fachlichen, anwenderspezifischen Informationen zu Entitätstypen, Attributen und Relationstypen in die konzeptionelle Schemaspezifikation übernommen werden.

LSE21.F.3 Integritätsbedingungen

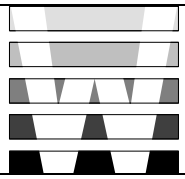
LSE21.F.3.1 Logisch
Es ist sichergestellt, daß bei der Transformation einer logischen Schemaspezifikation in eine konzeptionelle Schemaspezifikation die in der logischen Schemaspezifikation enthaltenen Integritätsbedingungen entweder in die Darstellung der konzeptionellen Schemaspezifikation oder in eine dazugehörige Beschreibung übernommen werden.

LSE21.F.3.2 Physisch
Es ist sichergestellt, daß bei der Transformation einer physischen Schemaspezifikation in eine konzeptionelle Schemaspezifikation die in den physischen Schemaspezifikationen enthaltenen Integritätsbedingungen entweder in die Darstellung der konzeptionellen Schemaspezifikation oder in eine dazugehörige Beschreibung übernommen werden.

LSE21.F.4 Transformation physischer in konzeptionelle Schemaspezifikationen
Es ist möglich, physische Schemaspezifikationen in konzeptionelle Schemaspezifikationen zu transformieren.

LSE21.F.5 Vollständigkeit der Transformation
Es ist sichergestellt, daß bei der Transformation einer physischen Schemaspezifikation in eine konzeptionelle Schemaspezifikation alle in der physischen Schemaspezifikation vorhandenen fachlichen, anwenderspezifischen Informationen zu Entitätstypen, Attributen und Relationstypen in die konzeptionelle Schemaspezifikation übernommen werden.

LSE21.F.6 Integration konzeptioneller Schemaspezifikationen
Es ist möglich, erstellte konzeptionelle Schemaspezifikationen zu übernehmen und diese mit bestehenden konzeptionellen Schemaspezifikationen zu einer konsistenten konzeptionellen Schemaspezifikation zu integrieren.



3.6.21.3.4 Sonstige Anforderungen

LSE21.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE21.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE21.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.22 Leistungseinheit LSE22 „Klassen-/Objektmodellierung unterstützen“

3.6.22.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivitäten (-Produkte)

- SE 1.1 Ist-Aufnahme/-Analyse durchführen
  („Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“)
- SE 1.2 Anwendungssystem beschreiben
  („Anwenderforderungen.*Grobe Systembeschreibung*“)
- SE 1.4 Randbedingungen definieren
  („Anwenderforderungen.*Randbedingungen*“)
- SE 1.5 System fachlich strukturieren
  („Anwenderforderungen.*Beschreibung der Funktionalität*“)
- SE 2.1 System technisch entwerfen
  („Systemarchitektur.*Lösungsvorschläge*“,
  „Systemarchitektur.*Darstellung der technischen Systemarchitektur*“,
  „Technische Anforderungen.*Gesamtfunktion des Elements*“,
  „Technische Anforderungen.*Technische Anforderungen an die Schnittstellen*“,
  „Schnittstellenübersicht.*Systemexterne Schnittstellen*“ und
  „Schnittstellenübersicht.*Systeminterne Schnittstellen*“)
- SE 2.5 Schnittstellen beschreiben
  („Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)
- SE 3.2 Anforderungen an die externen Schnittstellen der SW-/HW-Einheit präzisieren
  („Technische Anforderungen.*Technische Anforderungen an die Schnittstellen*“)
- SE 3.3 Anforderungen an die Funktionalität definieren
  („Technische Anforderungen.*Gesamtfunktion des Elements*“)
- SE 4.1-SW SW-Architektur entwerfen
  („SW-Architektur.*Lösungsvorschläge*“,
  „SW-Architektur.*Modularisierung/Datenbankentwurf*“,
  „SW-Architektur.*Schnittstellen*“ und
  „Schnittstellenübersicht.*Systeminterne Schnittstellen*“)
- SE 4.2-SW SW-interne und -externe Schnittstellen entwerfen
  („Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)
- SE 5.1-SW SW-Komponente/-Modul/Datenbank beschreiben
  („SW-Entwurf.*SW-Komponenten-/SW-Modul-Beschreibung*“ und
  „SW-Entwurf.*Datenbankbeschreibung*“ [bei objektorientierten Datenbanken])

Methode

- Klassen-/Objektmodellierung (KOM)

3.6.22.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Unterstützung der objektorientierten Methode „Klassen-/Objektmodellierung“ (KOM) gestellt.

3.6.22.3 Anforderungen

3.6.22.3.1 Anforderungen an die Schnittstellen

LSE22.S.1 Schnittstelle zu LOBV01 „Datenstrukturen und Daten der SEU erstellen, speichern, verwalten“
Die mit der Methode generierte Information wird in einem zentralen Repository gespeichert.

LSE22.S.2 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE22.S.3 Schnittstelle zu LSE23 „Subsystemmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE23 definierten logischen Gliederung des Systems zuzugreifen.

LSE22.S.4 Schnittstelle zu LSE24 „Moduldiagramme unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE24 definierten Modulstruktur zuzugreifen.

LSE22.S.5 Schnittstelle zu LSE25 „Prozeßdiagramme unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE25 definierten Prozeßstruktur zuzugreifen.

LSE22.S.6 Schnittstelle zu LSE26 „Use-Case-Modellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE26 definierten Use Cases zuzugreifen.

LSE22.S.7 Schnittstelle zu LSE27 „Zustandsmodellierung im objektorientierten Bereich unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE27 definierten Zuständen, Ereignissen und Aktionen zuzugreifen.

LSE22.S.8 Schnittstelle zu LSE28 „Interaktionsmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE28 definierten Objekten und Interaktionen zuzugreifen.

3.6.22.3.2 Anforderungen an die Methodenunterstützung

LSE22.M.1 Klassen-/Objektmodellierung (KOM)

LSE22.M.1.1 Symbole

LSE22.M.1.1.1 Klassen
Es steht ein Symbol zur Darstellung von Klassen zur Verfügung. Das Symbol erlaubt die getrennte Darstellung von zugehörigen Attributen und Operationen.

LSE22.M.1.1.2 Objekte
Es steht ein Symbol zur Darstellung von Objekten zur Verfügung.

LSE22.M.1.1.3 Utilities
Es steht ein Symbol zur Zusammenfassung von globalen Operationen, globalen Variablen und Bibliotheksfunktionen zur Verfügung.

LSE22.M.1.1.4 Beziehungen

LSE22.M.1.1.4.1 Vererbung
Es steht ein Symbol zur Darstellung von Vererbungsbeziehungen zur Verfügung.

LSE22.M.1.1.4.2 Assoziation
Es steht ein Symbol zur Darstellung von Assoziationsbeziehungen zur Verfügung.

LSE22.M.1.1.4.3 Aggregation
Es steht ein Symbol zur Darstellung von Aggregationsbeziehungen zur Verfügung.

LSE22.M.1.1.4.4 Objektbeziehungen
Es steht ein Symbol zur Darstellung von Beziehungen zwischen Objekten („Links“) zur Verfügung.

LSE22.M.1.1.4.5 Instantiierung
Es steht ein Symbol zur Darstellung von Instantiierungsbeziehungen zwischen Parametrierten Klassen und Instantiierten Klassen sowie zwischen Klassen und Objekten zur Verfügung.

Bei Instantiierten Klassen handelt es sich um normale Klassen, die aus Parametrierten Klassen durch die Angabe aktueller Parameter entstehen.

LSE22.M.1.2 Namen
Es ist möglich, für Klassen, Objekte und Beziehungen identifizierende Namen zu vergeben.

Um Namen eindeutig zu vergeben, ist es in Abhängigkeit vom Gültigkeitsbereich notwendig, Objektnamen um den zugehörigen Klassennamen bzw. Klassennamen um den zugehörigen Subsystemnamen zu ergänzen.

LSE22.M.1.3 Spezielle Klassentypen

LSE22.M.1.3.1 Parametrierte Klassen
Es ist möglich, Klassen als Parametrierte Klassen zu kennzeichnen und die formalen Parameter zu spezifizieren.

Parametrierte Klassen können zur Modellierung des beispielsweise aus C++ bekannten „Template“-Konzeptes verwendet werden.

LSE22.M.1.3.2 Metaklassen
Es ist möglich, Klassen als Metaklassen zu kennzeichnen.

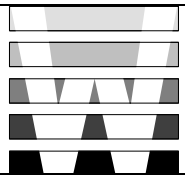
Mit diesem beispielsweise aus Smalltalk bekannten Konzept können Klassen zur Beschreibung von Klassen definiert werden. Instanzen von Metaklassen sind selbst Klassen.

LSE22.M.1.3.3 Abstrakte Klassen
Es ist möglich, Klassen als abstrakt zu kennzeichnen.
Abstrakte Klassen können zur Spezifikation von Abstraktionen verwendet werden, aus denen durch Vererbung konkrete Klassen abgeleitet werden können. Abstrakte Klassen können keine Instanzen haben.

LSE22.M.1.3.4 Assoziierte Klassen
Es ist möglich, Beziehungen durch die ergänzende Zuweisung von assoziierten Klassen zu verfeinern.

Die Beziehung kann dadurch feiner spezifiziert werden.

LSE22.M.1.4 Attribute



LSE22.M.1.4.1 Beschreibung
Es ist möglich, Attribute durch die Angabe eines Namens, eines Typs und eines Initialwertes zu spezifizieren. Zumindest die Namen müssen innerhalb von Klassensymbolen dargestellt werden können.

LSE22.M.1.4.2 Klassenattribute
Es ist möglich, Attribute als Klassenattribute zu kennzeichnen.

Klassenattribute sind klassenweit gültig und haben für alle Objekte einer Klasse den gleichen Wert.

LSE22.M.1.4.3 Objektattribute
Es ist möglich, für Objekte konkrete Attributwerte festzulegen. Die Objektattribute müssen innerhalb von Objektsymbolen dargestellt werden können.

LSE22.M.1.5 Operationen

LSE22.M.1.5.1 Beschreibung
Es ist möglich, Operationen durch die Angabe eines Namens, von Parametern und eines Rückgabewertes zu spezifizieren. Zumindest die Namen müssen innerhalb von Klassensymbolen dargestellt werden können.

LSE22.M.1.5.2 Spezifikationen von Operationen
Es ist möglich, Operationen durch die Angabe
– ihrer Aufgabenstellung bzw. Verantwortlichkeit,
– ihres Ein- und Ausgabeverhaltens,
– der Veränderung von Objekten und
– von Aussagen über den Systemzustand vor und nach der Ausführung feiner zu spezifizieren.

LSE22.M.1.6 Sichtbarkeit
Es ist möglich, die Sichtbarkeit von Attributen und Operationen zu kennzeichnen. Beispielsweise können gemäß C++ folgende Sichtbarkeitsstufen verwendet werden: „public“, „protected“, „private“.

LSE22.M.1.7 Verfeinerung von Beziehungen

LSE22.M.1.7.1 Kardinalität
Es ist möglich, Beziehungen durch die Angabe von Kardinalitäten feiner zu spezifizieren.

LSE22.M.1.7.2 Rollen
Es ist möglich, Beziehungen durch die Angabe von Rollen feiner zu spezifizieren.

Durch eine Rollenangabe kann die Beteiligung eines Objektes an einer Beziehung genauer ausgedrückt werden.

LSE22.M.1.7.3 Qualifikation
Es ist möglich, in Beziehungen die direkte Abhängigkeit zwischen Objekten durch die Angabe einer Qualifikation zu charakterisieren.

LSE22.M.1.8 Ableitungen
Es ist möglich, Attribute und Beziehungen als „abgeleitet“ zu kennzeichnen.

Abgeleitete Attribute kennzeichnen Informationen, die aus anderen Attributen berechnet werden können. Abgeleitete Beziehungen sind auch durch eine Kombination anderer Beziehungen repräsentiert.

LSE22.M.1.9 Bedingungen und Beschränkungen
Es ist möglich, Klassen und Beziehungen durch Bedingungen und Beschränkungen näher zu spezifizieren.

LSE22.M.1.10 Zusammensetzungen
Es ist möglich, bei der Modellierung von Klassen auch solche Attribute zu definieren, die Objekte und Beziehungen zwischen diesen repräsentieren.

LSE22.M.1.11 Schnittstellen
Es ist möglich, Schnittstellen zu Klassen und die Anwendung dieser Schnittstellen zu spezifizieren.

Durch eine Schnittstelle kann die Anwendung von Attributen und Operationen einer Klasse bzw. einer Gruppe von Klassen für einen bestimmten Problembereich benannt werden. Eine Schnittstelle kann die Sichtbarkeit bei Klassen weiter einschränken.

LSE22.M.1.12 Ergänzende Beschreibungen
Es ist möglich, Elemente wie Klassen, Objekte, Utilities, Beziehungen, Attribute und Operationen durch zusätzliche textliche Beschreibungen näher zu spezifizieren.

3.6.22.3.3 Anforderungen an die Funktionen

LSE22.F.1 Darstellung sprachspezifischer Informationen
Es ist möglich, Elemente der statischen Klassenstruktur für die Implementierung um zielsprachenspezifische Informationen zu ergänzen.

Durch gezielte Informationen kann die Implementierung in der Zielsprache näher spezifiziert werden. Eine automatische Codegenerierung kann unterstützt werden, wenn diese Informationen in einer für einen Generator verständlichen Form vorliegen.

LSE22.F.2 Codegenerierung
Es ist möglich, aus der mit der Methode KOM definierten statischen Klassenstruktur für die benötigte Zielsprache Quellcode zu generieren.

Aus der statischen Klassenstruktur können vollständige Spezifikationen und Rahmen für Rümpfe generiert werden.

LSE22.F.3 Codeerweiterung
Es ist möglich, daß Entwickler den generierten Quellcode erweitern und diese Erweiterungen bei einem erneuten Generierungslauf erhalten bleiben.

Erweiterungen sind zur Programmierung der Rümpfe erforderlich. Die vom Entwickler veränderbaren Quellcodeabschnitte können durch Kommentare gekennzeichnet werden.

LSE22.F.4 Reverse Engineering
Es ist möglich, aus existierendem Quellcode die statische Klassenstruktur in einer der Methode KOM entsprechenden Form zu generieren.

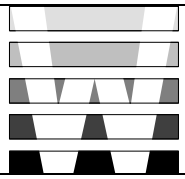
LSE22.F.5 Ausblenden
Es ist möglich, Teile von Klassen- bzw. Objektdiagrammen nach gewissen Kriterien auszublenden.

Beispielsweise ist es sinnvoll, in Klassendiagrammen die Attribut- bzw. Operationsblöcke ausblenden zu können, um bei komplexen Diagrammen die Übersichtlichkeit zu erhöhen.

LSE22.F.6 Aufteilung auf mehrere Diagramme
Es ist möglich, die mittels der Methode KOM definierten Informationen auf mehrere Diagramme gleichen Typs zu verteilen.

3.6.22.3.4 Sonstige Anforderungen

LSE22.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.



LSE22.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE22.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.23 Leistungseinheit LSE23 „Subsystemmodellierung unterstützen“

3.6.23.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivitäten (-Produkte)

- SE 1.2 Anwendungssystem beschreiben
  („Anwenderforderungen.*Grobe Systembeschreibung*“)
- SE 1.5 System fachlich strukturieren
  („Anwenderforderungen.*Fachliche Anforderungen*“)
- SE 2.1 System technisch entwerfen
  („Systemarchitektur.*Lösungsvorschläge*“,
  „Systemarchitektur.*Darstellung der technischen Systemarchitektur*“,
  „Technische Anforderungen.*Technische Anforderungen an die Schnittstellen*“ und
  „Schnittstellenübersicht.*Systeminterne Schnittstellen*“)
- SE 2.5 Schnittstellen beschreiben
  („Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)
- SE 4.1-SW SW-Architektur entwerfen
  („SW-Architektur.*Lösungsvorschläge*“,
  „SW-Architektur.*Modularisierung/Datenbankentwurf*“,
  „SW-Architektur.*Schnittstellen*“ und
  „Schnittstellenübersicht.*Systeminterne Schnittstellen*“)

Methode

- Subsystemmodellierung (SSM)

3.6.23.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Unterstützung der objektorientierten Methode „Subsystemmodellierung (SSM)“ gestellt.

3.6.23.3 Anforderungen

3.6.23.3.1 Anforderungen an die Schnittstellen

LSE23.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE23.S.2 Eingangs-Schnittstelle zu LSE22 „Klassen-/Objektmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE22 definierten Klassen-/Objektstruktur zuzugreifen.

LSE23.S.3 Eingangs-Schnittstelle zu LSE24 „Moduldiagramme unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE24 definierten physischen Gliederung des Systems zuzugreifen.

LSE23.S.4 Eingangs-Schnittstelle zu LSE29 „Formal spezifizieren“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE29 definierten Spezifikationen zuzugreifen.

LSE23.S.5 Eingangs-Schnittstelle zu LSE30 „Formal verifizieren“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE30 durchgeführten Verifikationen zuzugreifen.

LSE23.S.6 Eingangs-Schnittstelle zu LSE31 „Analyse verdeckter Kanäle“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE31 durchgeführten Analysen verdeckter Kanäle zuzugreifen.

3.6.23.3.2 Anforderungen an die Methodenunterstützung

LSE23.M.1 Subsystemmodellierung (SSM)

LSE23.M.1.1 Symbole

LSE23.M.1.1.1 Subsysteme
Es stehen Symbole zur Darstellung von Subsystemen zur Verfügung.

LSE23.M.1.1.2 Beziehungen
Es stehen Symbole zur Darstellung von Beziehungen zwischen Subsystemen zur Verfügung.

LSE23.M.1.2 Namen
Es ist möglich, für Subsysteme und Beziehungen zwischen Subsystemen identifizierende Namen zu vergeben.

LSE23.M.1.3 Ergänzende Beschreibungen
Es ist möglich, Subsysteme und Beziehungen zwischen Subsystemen durch zusätzliche textliche Beschreibungen näher zu spezifizieren.

LSE23.M.1.4 Verknüpfungen mit Klassen-/Objektinformationen
Es ist möglich, Subsysteme mit Klassen- und Objektinformationen zu verknüpfen.

LSE23.M.1.5 Verknüpfung von Klassen-/Objektinformationen
Es ist möglich, Klassen-/Objektinformationen aus verschiedenen Subsystemen miteinander zu verknüpfen.

LSE23.M.1.6 Hierarchische Zerlegung
Es ist möglich, Subsysteme hierarchisch in Subsysteme zu zerlegen.

LSE23.M.1.7 Navigieren

LSE23.M.1.7.1 Verfeinerte Ebene
Es ist möglich, aus einem Subsystemdiagramm zu einem Subsystemdiagramm einer verfeinerten Ebene zu navigieren.

LSE23.M.1.7.2 Nächsthöhere Ebene
Es ist möglich, aus einem Subsystemdiagramm zu einem Subsystemdiagramm der nächsthöheren Ebene zu navigieren.

LSE23.M.1.8 Subsystemdiagramme
Es ist möglich, die Darstellung von Subsystemen und ihren Beziehungen auf mehrere Diagramme zu verteilen.

3.6.23.3.3 Anforderungen an die Funktionen

keine

3.6.23.3.4 Sonstige Anforderungen

LSE23.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE23.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE23.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.24 Leistungseinheit LSE24 „Moduldiagramme unterstützen“

3.6.24.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivitäten (-Produkte)

– SE 4.1-SW SW-Architektur entwerfen
(„SW-Architektur.*Lösungsvorschläge*“,
„SW-Architektur.*Modularisierung/Datenbankentwurf*“ und
„Schnittstellenübersicht.*Systeminterne Schnittstellen*“)

– SE 5.1-SW SW-Komponente/-Modul/Datenbank beschreiben
(„SW-Entwurf.*SW-Komponenten-/SW-Modul-Beschreibung*“)

Methode

– Moduldiagramme (MODIAG)

3.6.24.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Unterstützung der objektorientierten Methode „Moduldiagramme“ (MODIAG) gestellt.

3.6.24.3 Anforderungen

3.6.24.3.1 Anforderungen an die Schnittstellen

LSE24.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE24.S.2 Eingangs-Schnittstelle zu LSE22 „Klassen-/Objektmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE22 definierten Klassen und Objekten zuzugreifen.

LSE24.S.3 Eingangs-Schnittstelle zu LSE23 „Subsystemmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE23 definierten logischen Gliederung des Systems zuzugreifen.

LSE24.S.4 Eingangs-Schnittstelle zu LSE25 „Prozeßdiagramme unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE25 definierten Prozeßstruktur zuzugreifen.

LSE24.S.5 Eingangs-Schnittstelle zu LSE29 „Formal spezifizieren“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE29 definierten Spezifikationen zuzugreifen.

LSE24.S.6 Eingangs-Schnittstelle zu LSE30 „Formal verifizieren“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE30 durchgeführten Verifikationen zuzugreifen.

LSE24.S.7 Eingangs-Schnittstelle zu LSE31 „Analyse verdeckter Kanäle“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE31 durchgeführten Analysen verdeckter Kanäle zuzugreifen.

3.6.24.3.2 Anforderungen an die Methodenunterstützung

LSE24.M.1 Moduldiagramme (MODIAG)

LSE24.M.1.1 Symbole

LSE24.M.1.1.1 Hauptprogramme
Es steht ein Symbol zur Darstellung von Hauptprogrammen zur Verfügung.

LSE24.M.1.1.2 Spezifikationen
Es steht ein Symbol zur Darstellung von Spezifikationen zur Verfügung.

LSE24.M.1.1.3 Rümpfe
Es steht ein Symbol zur Darstellung von Rümpfen zur Verfügung.

LSE24.M.1.1.3 Physische Subsysteme
Es steht ein Symbol zur Darstellung von physischen Subsystemen zur Verfügung.

LSE24.M.1.1.4 Compilierungsabhängigkeiten
Es steht ein Symbol zur Darstellung von Compilierungsabhängigkeiten zwischen den Modulen zur Verfügung.

LSE24.M.1.2 Zusätzliche Symbole
Es ist möglich, zur Darstellung anderer Compilierungseinheiten zusätzliche Symbole zu definieren.

LSE24.M.1.3 Namen
Es ist möglich, für Module identifizierende Namen zu vergeben.

LSE24.M.1.4 Ergänzende Modulbeschreibungen
Es ist möglich, Detailinformationen, die einzelne Module näher spezifizieren, in Textform in ergänzenden Modulbeschreibungen abzulegen.

LSE24.M.1.5 Zuordnung
Es ist möglich, den Modulen Objekte, freie Unterprogramme, Bibliotheken oder Zusammenfassungen von Klassen zuzuordnen, indem deren Bezeichner den Modulsymbolen zugeordnet werden oder indem die Zuordnung in separaten Klassen- oder Modulbeschreibungen aufgeführt wird.

3.6.24.3.3 Anforderungen an die Funktionen

keine

3.6.24.3.4 Sonstige Anforderungen

LSE24.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE24.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE24.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.25 Leistungseinheit LSE25 „Prozeßdiagramme unterstützen“

3.6.25.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivitäten (-Produkte)

- SE 1.1 Ist-Aufnahme/-Analyse durchführen
  („Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“)
- SE 1.2 Anwendungssystem beschreiben
  („Anwenderforderungen.*Grobe Systembeschreibung*“)
- SE 1.4 Randbedingungen definieren
  („Anwenderforderungen.*Randbedingungen*“)
- SE 2.1 System technisch entwerfen
  („Technische Anforderungen.*Gesamtfunktion des Elements*“,
  „Systemarchitektur.*Technischer Aufbau des Systems*“)
- SE 3.5 Anforderungen an Entwicklungs- und SWPÄ-Umgebung definieren
  („Technische Anforderungen.*Anforderungen an Entwicklungs- und SWPÄ-Umgebung*“)
- SE 4.1-SW SW-Architektur entwerfen
  („SW-Architektur.*Lösungsvorschläge*“ und
  „SW-Architektur.*Modularisierung/Datenbankentwurf*“)

Methode

- Prozeßdiagramme (PRODIAG)

3.6.25.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Unterstützung der objektorientierten Methode „Prozeßdiagramme“ (PRODIAG) gestellt.

3.6.25.3 Anforderungen

3.6.25.3.1 Anforderungen an die Schnittstellen

LSE25.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE25.S.2 Eingangs-Schnittstelle zu LSE22 „Klassen-/Objektmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE22 definierten Klassen und Objekten zuzugreifen.

LSE25.S.3 Eingangs-Schnittstelle zu LSE24 „Moduldiagramme unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE24 definierten Hauptprogrammen zuzugreifen.

LSE25.S.4 Eingangs-Schnittstelle zu LSE29 „Formal spezifizieren“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE29 definierten Spezifikationen zuzugreifen.

LSE25.S.5 Eingangs-Schnittstelle zu LSE30 „Formal verifizieren“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE30 durchgeführten Verifikationen zuzugreifen.

LSE25.S.6 Eingangs-Schnittstelle zu LSE31 „Analyse verdeckter Kanäle“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE31 durchgeführten Analysen verdeckter Kanäle zuzugreifen.

3.6.25.3.2 Anforderungen an die Methodenunterstützung

LSE25.M.1 Prozeßdiagramme (PRODIAG)

LSE25.M.1.1 Symbole

LSE25.M.1.1.1 Prozessoren
Es steht ein Symbol zur Darstellung von Hardwareeinheiten zur Verfügung, die Programme ausführen können.

LSE25.M.1.1.2 Geräte
Es steht ein Symbol zur Darstellung von Hardwareeinheiten zur Verfügung, die selbst keine Programme ausführen können.

LSE25.M.1.1.3 Verbindungen
Es steht ein Symbol zur Darstellung von physischen Kommunikationsverbindungen zwischen Prozessoren untereinander und von diesen zu Geräten zur Verfügung.

LSE25.M.1.2 Namen
Es ist möglich, für Prozessoren und Geräte identifizierende Namen zu vergeben.

LSE25.M.1.3 Zuordnung von Prozessen
Es ist möglich, Prozessorsymbole um eine Liste von Prozeßnamen zu ergänzen.

LSE25.M.1.4 Beschriftung von Verbindungen
Es ist möglich, Verbindungen durch eine Beschriftung zu erläutern.

LSE25.M.1.5 Ergänzende Beschreibungen
Es ist möglich, Detailinformationen zu Prozessoren, Geräten, Verbindungen und Prozessen in ergänzenden Beschreibungen abzulegen.

3.6.25.3.3 Anforderungen an die Funktionen

keine

3.6.25.3.4 Sonstige Anforderungen

LSE25.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE25.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE25.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.26 Leistungseinheit LSE26 „Use-Case-Modellierung unterstützen“

3.6.26.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivitäten (-Produkte)

- SE 1.1 Ist-Aufnahme/-Analyse durchführen („Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“)
- SE 1.2 Anwendungssystem beschreiben („Anwenderforderungen.*Grobe Systembeschreibung*“)
- SE 1.4 Randbedingungen definieren („Anwenderforderungen.*Randbedingungen*“)
- SE 1.5 System fachlich strukturieren („Anwenderforderungen.*Nutzung*“)

Methode

- Use-Case-Modellierung (UCM)

3.6.26.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Unterstützung der objektorientierten Methode „Use-Case-Modellierung“ (UCM) gestellt.

3.6.26.3 Anforderungen

3.6.26.3.1 Anforderungen an die Schnittstellen

LSE26.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE26.S.2 Eingangs-Schnittstelle zu LSE22 „Klassen-/Objektmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE22 definierten Klassen-/Objektstruktur zuzugreifen.

LSE26.S.3 Eingangs-Schnittstelle zu LSE27 „Zustandsmodellierung im objektorientierten Bereich unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE27 definierten dynamischen Modellen zuzugreifen.

LSE26.S.4 Eingangs-Schnittstelle zu LSE28 „Interaktionsmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE28 definierten Interaktionsmodellen zuzugreifen.

LSE26.S.5 Eingangs-Schnittstelle zu LSE29 „Formal spezifizieren“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE29 definierten Spezifikationen zuzugreifen.

LSE26.S.6 Eingangs-Schnittstelle zu LSE30 „Formal verifizieren“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE30 durchgeführten Verifikationen zuzugreifen.

LSE26.S.7 Eingangs-Schnittstelle zu LSE31 „Analyse verdeckter Kanäle“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE31 durchgeführten Analysen verdeckter Kanäle zuzugreifen.

3.6.26.3.2 Anforderungen an die Methodenunterstützung

LSE26.M.1 Use-Case-Modellierung (UCM)

LSE26.M.1.1 Symbole

LSE26.M.1.1.1 Bedienungseinheiten
Es steht ein Symbol zur Darstellung von Bedienungseinheiten („Aktoren“) zur Verfügung.

LSE26.M.1.1.2 Use Cases
Es steht ein Symbol zur Darstellung von Use Cases zur Verfügung.

LSE26.M.1.1.3 Systeme
Es steht ein Symbol zur Darstellung von Systemen als fachliche Einheiten zur Verfügung.

LSE26.M.1.1.4 Beziehungen
Es stehen Symbole zur Darstellung der Beziehungen zwischen Bedienungseinheiten und Use Cases sowie zwischen Use Cases untereinander zur Verfügung.

Die Beziehungen zwischen Use Cases können dahingehend gekennzeichnet werden, ob es sich um eine optionale (Extend-Beziehung) oder um eine obligatorische (Uses-Beziehung) Ergänzung handelt.

LSE26.M.1.2 Namen
Es ist möglich, für Bedienungseinheiten, Use Cases und Beziehungen identifizierende Namen zu vergeben.

LSE26.M.1.3 Use-Case-Diagramme
Es ist möglich, die Use Cases, die eine fachliche Einheit bilden, innerhalb eines Systemsymbols darzustellen. Die zu den Use Cases in Beziehung stehenden Bedienungseinheiten können außerhalb der Systemsymbole angeordnet werden und greifen von außen auf die in dieser Einheit liegenden Use Cases zu.

LSE26.M.1.4 Spezifikation von Use Cases
Es ist möglich, Use Cases in Textform zu spezifizieren.

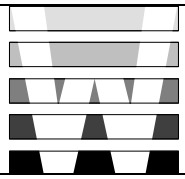
LSE26.M.1.5 Ergänzende Beschreibungen
Es ist möglich, Bedienungseinheiten und Beziehungen durch zusätzliche Textbeschreibungen näher zu spezifizieren.

3.6.26.3.3 Anforderungen an die Funktionen

keine

3.6.26.3.4 Sonstige Anforderungen

LSE26.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.



LSE26.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE26.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.27 Leistungseinheit LSE27 „Zustandsmodellierung im objektorientierten Bereich unterstützen“

3.6.27.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivitäten (-Produkte)

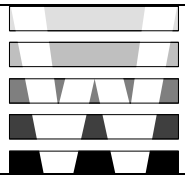
- SE 1.1 Ist-Aufnahme/-Analyse durchführen
  („Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“)
- SE 1.2 Anwendungssystem beschreiben
  („Anwenderforderungen.*Grobe Systembeschreibung*“)
- SE 1.4 Randbedingungen definieren
  („Anwenderforderungen.*Randbedingungen*“)
- SE 1.5 System fachlich strukturieren
  („Anwenderforderungen.*Beschreibung der Funktionalität*“)
- SE 2.5 Schnittstellen beschreiben
  („Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)
- SE 3.2 Anforderungen an die externen Schnittstellen der SW-/HW-Einheit präzisieren
  („Technische Anforderungen.*Technische Anforderungen an die Schnittstellen*“)
- SE 3.3 Anforderungen an die Funktionalität definieren
  („Technische Anforderungen.*Gesamtfunktion des Elements*“)
- SE 4.2-SW SW-interne und -externe Schnittstellen entwerfen
  („Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)
- SE 5.1-SW SW-Komponente/-Modul/Datenbank beschreiben
  („SW-Entwurf.*Realisierung der SW-Komponente/des SW-Moduls*“)

Methode

- Zustandsmodellierung im objektorientierten Bereich (ZUSTO)

3.6.27.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Unterstützung der Methode „Zustandsmodellierung im objektorientierten Bereich (ZUSTO)“ gestellt.



3.6.27.3 Anforderungen

3.6.27.3.1 Anforderungen an die Schnittstellen

LSE27.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE27.S.2 Eingangs-Schnittstelle zu LSE22 „Klassen-/Objektmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE22 definierten Klassen-/Objektstruktur zuzugreifen.

LSE27.S.3 Eingangs-Schnittstelle zu LSE26 „Use-Case-Modellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE26 definierten Use Cases zuzugreifen.

LSE27.S.4 Eingangs-Schnittstelle zu LSE28 „Interaktionsmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE28 definierten Interaktionsmodellen zuzugreifen.

LSE27.S.5 Eingangs-Schnittstelle zu LSE29 „Formal spezifizieren“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE29 definierten Spezifikationen zuzugreifen.

LSE27.S.6 Eingangs-Schnittstelle zu LSE30 „Formal verifizieren“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE30 durchgeführten Verifikationen zuzugreifen.

LSE27.S.7 Eingangs-Schnittstelle zu LSE31 „Analyse verdeckter Kanäle“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE31 durchgeführten Analysen verdeckter Kanäle zuzugreifen.

3.6.27.3.2 Anforderungen an die Methodenunterstützung

LSE27.M.1 Zustandsmodellierung im objektorientierten Bereich (ZUSTO)

LSE27.M.1.1 Symbole

LSE27.M.1.1.1 Zustände
Es steht ein Symbol zur Darstellung von Zuständen („States“) zur Verfügung.

LSE27.M.1.1.2 Anfangszustände
Es steht ein Symbol zur Darstellung von Anfangszuständen zur Verfügung.

LSE27.M.1.1.3 Schlußzustände
Es steht ein Symbol zur Darstellung von Schlußzuständen zur Verfügung.

LSE27.M.1.1.4 Zustandsübergänge
Es steht ein Symbol zur Darstellung von Zustandsübergängen („Transitions“) zur Verfügung.

LSE27.M.1.2 Ereignisse
Es ist möglich, den Zustandsübergängen Ereignisse („Events“) zuzuordnen, die die Übergänge in neue Zustände bewirken.

LSE27.M.1.3 Namen
Es ist möglich, Zuständen und Ereignissen identifizierende Namen zu geben.

LSE27.M.1.4 Attribute
Es ist möglich, Ereignisse mit Attributen zu versehen, die die Übergabe von Daten an eine das Ereignis realisierende Funktion repräsentieren.

LSE27.M.1.5 Bedingungen
Es ist möglich, den Zustandsübergängen Bedingungen zuzuordnen, die die Übergänge in neue Zustände nur dann erlauben, wenn die Bedingungen erfüllt sind und die betreffenden Ereignisse eintreten.

LSE27.M.1.6 Aktionen

LSE27.M.1.6.1 Aktionen bei Zustandsübergängen
Es ist möglich, Zustandsübergängen Aktionen („Actions“) zuzuordnen, die sofort auszuführen sind, wenn die Zustandsübergänge erfolgen.

Der Zeitbedarf dieser Aktionen ist vernachlässigbar.

LSE27.M.1.6.2 Eintritts- und Austrittsaktionen
Es ist möglich, Zuständen Aktionen („Actions“) zuzuordnen, die beim Eintritt in Zustände bzw. beim Austritt aus Zuständen auszuführen sind.

Der Zeitbedarf dieser Aktionen ist vernachlässigbar.

LSE27.M.1.7 Aktivitäten
Es ist möglich, den Zuständen Aktivitäten („Activities“) zuzuordnen, die bei Erreichen des Zustandes auszuführen sind.

Aktivitäten erfordern eine zu berücksichtigende Ausführungszeit.

LSE27.M.1.8 Hierarchie

LSE27.M.1.8.1 Hierarchische Zerlegung
Es ist möglich, die in den Zustandsdiagrammen verwendeten Zustände hierarchisch zu verfeinern.

LSE27.M.1.8.2 Automatisches Anzeigen
Ein- und ausgehende Zustandsübergänge eines Zustands werden in Diagrammen auf der nächsttieferen Hierarchieebene automatisch angezeigt.

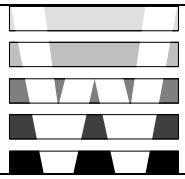
LSE27.M.1.9 Öffnen von Zustandsdiagrammen

LSE27.M.1.9.1 Nächsttiefere Ebene
Es ist möglich, aus einem Zustandsdiagramm ein Zustandsdiagramm der nächsttieferen Ebene in der Baumstruktur zu öffnen.

LSE27.M.1.9.2 Nächsthöhere Ebene
Es ist möglich, aus einem Zustandsdiagramm ein Zustandsdiagramm der nächsthöheren Ebene in der Baumstruktur zu öffnen.

LSE27.M.1.10 Zusammenhänge zwischen Zustandsdiagrammen verschiedener Klassen
Es ist möglich, das Senden von Ereignissen an Objekte anderer Klassen zu spezifizieren.

LSE27.M.1.11 Erzeugen und Löschen von Objekten
Es ist möglich, Ereignisse zu spezifizieren, die das Erzeugen bzw. Löschen von Objekten bewirken.



LSE27.M.1.12 Nebenläufigkeiten

LSE27.M.1.12.1 Nebenläufigkeiten in Zuständen
Es ist möglich, in komplexen Zuständen das nebenläufige Auftreten von Zustands- und Ereignisfolgen zu kennzeichnen.

LSE27.M.1.12.2 Splitten von Zustandsübergängen
Es ist möglich, zu kennzeichnen, daß ein Zustandsübergang den Eintritt in mehrere nebenläufige Zustandsfolgen bewirkt.

LSE27.M.1.12.3 Synchronisation von Zustandsübergängen
Es ist möglich, zu kennzeichnen, daß nur das Auftreten von Zustandsübergängen aus mehreren nebenläufigen Zustandsfolgen den Übergang in einen neuen Zustand bewirkt.

LSE27.M.1.13 Ergänzende Beschreibungen
Es ist möglich, Zustände, Zustandsübergänge, Ereignisse, Bedingungen, Aktionen und Aktivitäten durch zusätzliche Textbeschreibungen näher zu spezifizieren.

3.6.27.3.3 Anforderungen an die Funktionen

keine

3.6.27.3.4 Sonstige Anforderungen

LSE27.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE27.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE27.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.28 Leistungseinheit LSE28 „Interaktionsmodellierung unterstützen“

3.6.28.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivitäten (-Produkte)

- SE 1.1 Ist-Aufnahme/-Analyse durchführen
  („Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“)
- SE 1.2 Anwendungssystem beschreiben
  („Anwenderforderungen.*Grobe Systembeschreibung*“)
- SE 1.4 Randbedingungen definieren
  („Anwenderforderungen.*Randbedingungen*“)
- SE 1.5 System fachlich strukturieren
  („Anwenderforderungen.*Beschreibung der Funktionalität*“)
- SE 2.1 System technisch entwerfen
  („Systemarchitektur.*Erläuterung der Zusammenarbeit technischer Elemente*“)
- SE 2.5 Schnittstellen beschreiben
  („Schnittstellenbeschreibung.Beschreibung der Schnittstellen“)
- SE 3.2 Anforderungen an die externen Schnittstellen der SW-/HW-Einheit präzisieren
  („Technische Anforderungen.*Technische Anforderungen an die Schnittstellen*“)
- SE 3.3 Anforderungen an die Funktionalität definieren
  („Technische Anforderungen.*Gesamtfunktion des Elements*“)
- SE 4.1-SW SW-Architektur entwerfen
  („SW-Architektur.*Dynamisches Ablaufmodell*“)
- SE 4.2-SW SW-interne und -externe Schnittstellen entwerfen
  („Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)
- SE 5.1-SW SW-Komponente/-Modul/Datenbank beschreiben
  („SW-Entwurf.*SW-Komponenten-/SW-Modul-Beschreibung*“)

Methode

- Interaktionsmodellierung (IAM)

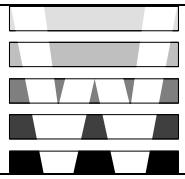
3.6.28.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Unterstützung der objektorientierten Methode „Interaktionsmodellierung“ (IAM) gestellt.

3.6.28.3 Anforderungen

3.6.28.3.1 Anforderungen an die Schnittstellen

LSE28.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist



für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE28.S.2 Eingangs-Schnittstelle zu LSE22 „Klassen-/Objektmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE22 definierten Operationen und Objekten zuzugreifen.

LSE28.S.3 Eingangs-Schnittstelle zu LSE26 „Use-Case-Modellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE26 definierten Use Cases zuzugreifen.

LSE28.S.4 Eingangs-Schnittstelle zu LSE27 „Zustandsmodellierung im objektorientierten Bereich unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu dem mit LSE27 definierten dynamischen Verhalten zuzugreifen.

LSE28.S.5 Eingangs-Schnittstelle zu LSE29 „Formal spezifizieren“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE29 definierten Spezifikationen zuzugreifen.

LSE28.S.6 Eingangs-Schnittstelle zu LSE30 „Formal verifizieren“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE30 durchgeführten Verifikationen zuzugreifen.

LSE28.S.7 Eingangs-Schnittstelle zu LSE31 „Analyse verdeckter Kanäle“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE31 durchgeführten Analysen verdeckter Kanäle zuzugreifen.

3.6.28.3.2 Anforderungen an die Methodenunterstützung

LSE28.M.1 Interaktionsmodellierung (IAM)

LSE28.M.1.1 Zeitliniendiagramme

LSE28.M.1.1.1 Symbole

LSE28.M.1.1.1.1 Objektlinien
Es ist möglich, Objekte mit ihrer Lebensdauer als vertikale Linien darzustellen. Das obere Ende der Objektlinie markiert den Zeitpunkt, an dem das Objekt erzeugt wird, das untere Ende den Zeitpunkt, an dem es gelöscht wird.

LSE28.M.1.1.1.2 Horizontale Interaktionspfeile
Es ist möglich, Interaktionen als horizontale Pfeile darzustellen. Interaktionspfeile symbolisieren Ereignisse oder Nachrichten und verlaufen von einer Objektlinie zu einer anderen.

LSE28.M.1.1.1.3 Abwärts gerichtete Interaktionspfeile
Es ist möglich, Interaktionen als abwärts gerichtete Pfeile darzustellen. Diese Interaktionspfeile symbolisieren einen Zeitbedarf für ein Ereignis und verlaufen von einer Objektlinie zu einer anderen.

LSE28.M.1.1.1.4 Zeitmarken
Zeitmarken können in Form von Zeichenketten an den Objektlinien angeordnet werden.

LSE28.M.1.1.1.5 Zeitbeschränkungen
Zeitbeschränkungen können symbolisiert und durch Boolesche Ausdrücke spezifiziert werden.

LSE28.M.1.1.2 Namen

LSE28.M.1.1.2.1 Objektnamen
Es ist möglich, für Objekte identifizierende Namen zu vergeben und diese im Diagramm oberhalb der Objektlinien anzuzeigen.

LSE28.M.1.1.2.2 Interaktionsnamen
Es ist möglich, für Interaktionen identifizierende Namen zu vergeben.
Mit den Namen sind die Ereignisse bzw. die Nachrichten zu bezeichnen, die die Interaktionen repräsentieren. Nachrichten können mit den Namen der Operationen (mit Parametern) bezeichnet werden, durch die sie zu realisieren sind.

LSE28.M.1.1.3 Reihenfolge von Interaktionen
Es ist möglich, die Reihenfolge der Interaktionen durch die Anordnung ihrer Symbole auszudrücken. Die Reihenfolge ist im Diagramm von oben nach unten auszudrücken.

LSE28.M.1.1.4 Bedingungen
Es ist möglich, Interaktionen mit Zeit-, Ablauf- und Synchronisationsbedingungen zu versehen.

LSE28.M.1.1.5 Verzweigungen, Iterationen und Rekursion
Es ist möglich, für Interaktionen alternative, iterative und rekursive Anwendungen zu modellieren.

LSE28.M.1.1.6 Erzeugen und Löschen von Objekten
Es ist möglich, das Erzeugen und Löschen von Objekten zu modellieren.

LSE28.M.1.1.7 Synchroner und asynchroner Nachrichtenfluß
Es ist möglich, synchrone und asynchrone Nachrichtenflüsse zu modellieren.

LSE28.M.1.2 Interaktionsgraphen

LSE28.M.1.2.1 Symbole

LSE28.M.1.2.1.1 Objekte
Es steht ein Symbol zur Darstellung von Objekten zur Verfügung.

LSE28.M.1.2.1.2 Objektbeziehungen
Es steht ein Symbol zur Darstellung von Beziehungen zwischen Objekten („Links“) zur Verfügung.

LSE28.M.1.2.1.3 Implementierungssymbol
Es steht ein Symbol zur Darstellung unterschiedlicher Realisierungsformen für eine Objektbeziehung zur Verfügung.

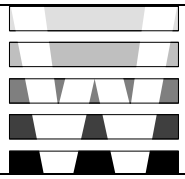
Eine Realisierungsform charakterisiert die Realisierung der Beziehung bezüglich eines Objektes. Für eine Beziehung können daher für jedes beteiligte Objekt unterschiedliche Realisierungsformen angegeben werden. Beispiele für Realisierungsformen sind: Assoziationsbeziehung, ein Objekt ist Bestandteil des anderen, Globale Variable, Lokale Variable und Prozedurparameter.

LSE28.M.1.2.1.4 Richtungspfeile

LSE28.M.1.2.1.4.1 Synchroner Nachrichtenfluß
Es steht ein Symbol zur Darstellung eines synchronen Nachrichtenflusses zur Verfügung.

Ein Objekt sendet eine Nachricht an ein anderes Objekt und wartet auf dessen Reaktion (prozeduraler Steuerfluß).

LSE28.M.1.2.1.4.2 Asynchroner Nachrichtenfluß
Es steht ein Symbol zur Darstellung eines asynchronen Nachrichtenflusses zur Verfügung.



Ein Objekt sendet eine Nachricht an ein anderes Objekt und wartet nicht auf eine Reaktion.

LSE28.M.1.2.2 Namen
Es ist möglich, für Objekte und Nachrichten identifizierende Namen zu vergeben. Um Namen eindeutig zu vergeben, ist es gegebenenfalls in Abhängigkeit vom Gültigkeitsbereich notwendig, Objektnamen um den zugehörigen Klassennamen bzw. Klassennamen um den zugehörigen Subsystemnamen zu ergänzen.

LSE28.M.1.2.3 Rollen
Es ist möglich, Beziehungen durch die Angabe von Rollen feiner zu spezifizieren. Durch eine Rollenangabe kann die Beteiligung eines Objektes an einer Beziehung genauer ausgedrückt werden.

LSE28.M.1.2.4 Erzeugen und Löschen
Es ist möglich, Objekte und Beziehungen dahingehend zu kennzeichnen, daß sie durch das Senden von Nachrichten erzeugt bzw. gelöscht werden.

LSE28.M.1.2.5 Nachrichten

LSE28.M.1.2.5.1 Sequenznummer
Es ist möglich, Nachrichten mit Sequenznummern zu versehen und damit ihre Reihenfolge festzulegen.

LSE28.M.1.2.5.2 Argumentliste und Rückgabewert
Es ist möglich, für Nachrichten eine Liste von Argumenten und einen Rückgabewert zu spezifizieren.

LSE28.M.1.2.5.3 Zuordnung
Es ist möglich, einer Objektbeziehung eine Nachricht oder mehrere Nachrichten zuzuordnen.

LSE28.M.1.2.5.4 Nachrichtenfluß
Es ist möglich, den Nachrichtenfluß durch Richtungspfeile näher zu spezifizieren.

LSE28.M.1.2.5.5 Nebenläufigkeiten

LSE28.M.1.2.5.5.1 Kennzeichnung
Es ist möglich, die nebenläufige Bearbeitung von Nachrichten auszudrücken.

LSE28.M.1.2.5.5.2 Bedingungen an die Verarbeitungsreihenfolge
Es ist möglich, bei der Kennzeichnung nebenläufiger Nachrichten zu vermerken, daß vor ihrer Bearbeitung andere Nachrichten bearbeitet sein müssen.

LSE28.M.1.2.5.5.3 Steuerflüsse für parallel aktive Objekte
Es ist möglich, die Steuerflüsse mehrerer aktiver Objekte in einem Interaktionsgraphen darzustellen und den Nachrichtenfluß zwischen ihnen auszudrükken.

LSE28.M.1.2.5.6 Iterationen
Es ist möglich, durch eine Iterationsangabe zu kennzeichnen, daß sequentiell oder nebenläufig mehrere Nachrichten der gleichen Form gesendet werden.

LSE28.M.1.2.5.7 Verzweigungen
Es ist möglich, durch die Angabe von Bedingungen die alternative Ausführung von Interaktionen zu modellieren.

LSE28.M.1.2.5.8 Rekursion
Es ist möglich, die rekursive Ausführung von Interaktionen zu modellieren.

3.6.28.3.3 Anforderungen an die Funktionen

keine

3.6.28.3.4 Sonstige Anforderungen

LSE28.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE28.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE28.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.6.29 Leistungseinheit LSE29 „Formal spezifizieren“

3.6.29.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– SE 2.5 Schnittstellen beschreiben
(„Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)

– SE 4.1-SW SW-Architektur entwerfen
(„SW-Architektur.*Lösungsvorschläge*“,
„SW-Architektur.*Modularisierung/Datenbankentwurf*“,
„SW-Architektur.*Schnittstellen*“ und
„Schnittstellenübersicht.*Systeminterne Schnittstellen*“)

– SE 4.2-SW SW-interne und -externe Schnittstellen entwerfen
(„Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)

– SE 5.1-SW SW-Komponente/-Modul/Datenbank beschreiben
(„SW-Entwurf.S*W-Komponenten-/SW-Modul-Beschreibung*“ und
„SW-Entwurf.*Datenbankbeschreibung*“ [bei objektorientierten Datenbanken])

Methode

– Formale Spezifikation (FS)

3.6.29.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Unterstützung der „Formalen Spezifikation“ (FS) gestellt. An dieses Werkzeug stellen die „ITSEC“ je nach Evaluationsklasse unterschiedliche Anforderungen. Die Anforderungen sind in dem Dokument „Kriterien für die Entwicklung, Realisierung und Zulassung von Werkzeugen zur formalen Spezifikation und Verifikation“ beschrieben.

Eine formale Spezifikation ist für das formale Sicherheitsmodell erforderlich, das ab der Evaluationsstufe E4 gemäß ITSEC gefordert ist. Sie wird auch für den Architekturentwurf in der Stufe E6 gemäß ITSEC gefordert.

3.6.29.3 Anforderungen

3.6.29.3.1 Anforderungen an die Schnittstellen

LSE29.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE29.S.2 Eingangs-Schnittstelle zu LSE23 „Subsystemmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE23 definierten logischen Gliederung des Systems zuzugreifen.

LSE29.S.3 Eingangs-Schnittstelle zu LSE24 „Moduldiagramme unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE24 definierten Modulstruktur zuzugreifen.

LSE29.S.4 Eingangs-Schnittstelle zu LSE25 „Prozeßdiagramme unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE25 definierten Prozeßstruktur zuzugreifen.

LSE29.S.5 Eingangs-Schnittstelle zu LSE26 „Use-Case-Modellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE26 definierten Use Cases zuzugreifen.

LSE29.S.6 Eingangs-Schnittstelle zu LSE27 „Zustandsmodellierung im objektorientierten Bereich unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE27 definierten Zuständen, Ereignissen und Aktionen zuzugreifen.

LSE29.S.7 Eingangs-Schnittstelle zu LSE28 „Interaktionsmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE28 definierten Objekten und Interaktionen zuzugreifen.

LSE29.S.8 Ausgangs-Schnittstelle zu LSE30 „Formal verifizieren“
Es ist möglich, die erstellte formale Spezifikation von dem Werkzeug zur formalen Verifikation als Input aufnehmen zu lassen.

LSE29.S.9 Ausgangs-Schnittstelle zu LSE31 „Analyse verdeckter Kanäle“
Es ist möglich, die erstellte formale Spezifikation von dem Werkzeug zur Analyse verdeckter Kanäle als Input aufnehmen zu lassen.

3.6.29.3.2 Anforderungen an die Methodenunterstützung

LSE29.M.1 Formale Spezifikation (FS)

LSE29.M.1.1 Formale Spezifikationssprache
Das Werkzeug ist auf die verwendete(n) Spezifikationssprache(n) abgestimmt.

LSE29.M.1.1.1 Syntax
Die Spezifikationssprache besitzt eine formal definierte Syntax.

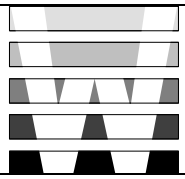
LSE29.M.1.1.2 Semantik
Die Spezifikationssprache besitzt eine formal definierte Semantik. Wird ein Sprachkonstrukt mit verschiedenen Bedeutungen verwendet, so muß die Semantik eindeutig aus dem Kontext festgelegt sein.

LSE29.M.1.1.3 Ausdrucksfähigkeit
Die Spezifikationssprache ist auf die Ausdrucksfähigkeit anderer eingesetzter Sprachen, insbesondere der Zielsprache, abgestimmt.

LSE29.M.1.1.4 Konzepte zur Strukturierung

LSE29.M.1.1.4.1 Hierarchisch angeordnete Spezifikationsebenen
Es ist möglich, hierarchisch angeordnete Spezifikationsebenen zur Strukturierung einzusetzen.

LSE29.M.1.1.4.2 Modulare Aufteilung der Spezifikation
Es ist möglich, eine modulare Aufteilung der Spezifikation auf jeder Spezifikationsebene zur Strukturierung einzusetzen.



LSE29.M.1.1.5 Sprachkonstrukte zur Spezifikation von Prozessen

LSE29.M.1.1.5.1 Sequentielle und nebenläufige Prozesse
Es ist möglich, sequentielle und nebenläufige Prozesse zu spezifizieren.

LSE29.M.1.1.5.2 Prozeßkommunikation
Es ist möglich, die Kommunikation zwischen Prozessen zu spezifizieren.

LSE29.M.1.1.6 Verständlichkeit
Die formale Spezifikationssprache verwendet geläufige mathematische Symbole und Schreibweisen.

LSE29.M.1.1.7 Strenges Typkonzept
Die Spezifikationssprache verfolgt ein strenges Typkonzept.

3.6.29.3.3 Anforderungen an die Funktionen

LSE29.F.1 Rapid Prototyping
Es ist möglich, Teile der mit der Methode FS definierten Spezifikation auszuführen oder für die benötigte Zielsprache Quellcode zu generieren.

LSE29.F.2 Syntaxprüfung
Es ist möglich, daß das Werkzeug die Syntax der Spezifikation überprüft.

LSE29.F.3 Typprüfung
Es ist möglich, daß das Werkzeug die Spezifikation auf Typfehler und statische Semantik (Kontextbedingungen) prüft.

LSE29.F.4 Editor

LSE29.F.4.1 Eingabe von Formeln
Es gibt nutzerfreundliche Hilfsmittel zur Eingabe von Formeln der formalen Spezifikationssprache(n).

Beispielsweise sollte ein einheitlicher, leicht handhabbarer Editor zur Verfügung stehen.

LSE29.F.4.2 Strukturierte Darstellung
Für Ausgaben des Werkzeugs ist deutlich, in welchem Kontext sie stehen. Sie müssen gut lesbar, leicht verständlich und präzise sein.

LSE29.F.5 Formelsammlungen
Es gibt Datenbanken (Formelsammlungen), in denen Basis-Bausteine wie formal spezifizierte Module, elementare Datenstrukturen mit entsprechenden Theorien, Theoreme und Lemmata abgelegt sind, auf die die Komponenten des Werkzeugs zugreifen können.

3.6.29.3.4 Sonstige Anforderungen

LSE29.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE29.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen. Die Syntax dieser Sprache muß formal definiert sein.

LSE29.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LSE29.So.4 Einheitliche Nutzerschnittstelle

LSE29.So.4.1 Die Oberfläche ist für alle Teile des Spezifikationswerkzeugs einheitlich.

LSE29.So.4.2 Die Oberfläche des Spezifikationswerkzeugs und des Verifikationswerkzeugs sind gleich.

LSE29.So.5 Effizienz

Mit dem Werkzeug sind – gemessen am Stand der Technik – Entwicklungen in akzeptabler Zeit und mit akzeptablem Speicheraufwand durchführbar.

LSE29.So.6 Installation

LSE29.So.6.1 Verfahren

Es ist ein Verfahren zur Installation des Spezifikationswerkzeugs festgelegt und mit sämtlichen Parametern dokumentiert.

LSE29.So.6.2 Konfigurationsmöglichkeiten

Alle Konfigurationsmöglichkeiten für den Werkzeug-Start durch den Nutzer und die entsprechenden Parameter sind mit ihren Auswirkungen auf das Gesamtverhalten des Werkzeugs dokumentiert.

LSE29.So.6.3 Authentizität des Binärcodes

Die Verfahren zur Sicherstellung der Authentizität des Binärcodes, zur Installation und zum Start eines Werkzeugs werden beschrieben . Das Vorgehen bei der Wartung des Werkzeugs mit der entsprechenden Versionskontrolle ist dokumentiert.

LSE29.So.7 Dokumentation

LSE29.So.7.1 Nutzerdokumentation

Die Nutzerdokumentation wendet sich an einen Nutzer mit Kenntnissen zur formalen Spezifikation und beinhaltet alle zur Arbeit mit dem Werkzeug nötigen Informationen.

LSE29.So.7.2 Sonstige Dokumentation

Die verwendete Methodik und mathematischen Grundlagen werden präzise beschrieben.

LSE29.So.8 Zulassung beim BSI

Für Entwicklungen nach E5 oder E6 gemäß ITSEC ist das Werkzeug vom BSI zugelassen.

3.6.30 Leistungseinheit LSE30 „Formal verifizieren“

3.6.30.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- SE 2.5 Schnittstellen beschreiben
  („Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)
- SE 4.1-SW SW-Architektur entwerfen
  („SW-Architektur.*Lösungsvorschläge*“,
  „SW-Architektur.*Modularisierung/Datenbankentwurf*“,
  „SW-Architektur.*Schnittstellen*“ und
  „Schnittstellenübersicht.*Systeminterne Schnittstellen*“)
- SE 4.2-SW SW-interne und -externe Schnittstellen entwerfen
  („Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)
- SE 5.1-SW SW-Komponente/-Modul/Datenbank beschreiben
  („SW-Entwurf.*SW-Komponenten-/SW-Modul-Beschreibung*“ und
  „SW-Entwurf.*Datenbankbeschreibung*“ [bei objektorientierten Datenbanken])
- SE 6.1-SW SW-Module codieren
  („SW-Modul“)
- SE 6.2-SW Datenbank realisieren
  („Datenbank“)
- SE 6.3-SW Selbstprüfung des SW-Moduls/der Datenbank durchführen
  („Implementierungsdokumente (SW-Modul)“,
  „Implementierungsdokumente (Datenbank)“,
  „SW-Modul“ und
  „Datenbank“)

Methode

- Designverifikation (DVER)
- Programmverifikation (PVER)

3.6.30.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Unterstützung der „Formalen Verifikation“ gestellt, daß eine Spezifikation eine Verfeinerung einer anderen Spezifikation ist (DVER) bzw. daß ein Programm eine Spezifkation erfüllt (PVER). An dieses Werkzeug stellen die „ITSEC“ je nach Evaluationsklasse unterschiedliche Anforderungen. Die Anforderungen sind in dem Dokument „Kriterien für die Entwicklung, Realisierung und Zulassung von Werkzeugen zur formalen Spezifikation und Verifikation“ beschrieben. Die ITSEC fordern aber lediglich für das formale Sicherheitsmodell (ab der Evaluationsstufe E4) eine formale Verifikation.

3.6.30.3 Anforderungen

3.6.30.3.1 Anforderungen an die Schnittstellen

LSE30.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE30.S.2 Eingangs-Schnittstelle zu LSE23 „Subsystemmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE23 definierten logischen Gliederung des Systems zuzugreifen.

LSE30.S.3 Eingangs-Schnittstelle zu LSE24 „Moduldiagramme unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE24 definierten Modulstruktur zuzugreifen.

LSE30.S.4 Eingangs-Schnittstelle zu LSE25 „Prozeßdiagramme unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE25 definierten Prozeßstruktur zuzugreifen.

LSE30.S.5 Eingangs-Schnittstelle zu LSE26 „Use-Case-Modellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE26 definierten Use Cases zuzugreifen.

LSE30.S.6 Eingangs-Schnittstelle zu LSE27 „Zustandsmodellierung im objektorientierten Bereich unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE27 definierten Zuständen, Ereignissen und Aktionen zuzugreifen.

LSE30.S.7 Eingangs-Schnittstelle zu LSE28 „Interaktionsmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE28 definierten Objekten und Interaktionen zuzugreifen.

LSE30.S.8 Eingangs-Schnittstelle zu LSE29 „Formal spezifizieren“
Es ist möglich, die erstellte formale Spezifikation von dem Werkzeug zur formalen Verifikation weiterverarbeiten zu lassen.

LSE30.S.9 Eingangs-Schnittstelle zu LSE31 „Analyse verdeckter Kanäle“
Es ist möglich, die im Werkzeug zur Analyse verdeckter Kanäle erstellten Verifikationsbedingungen im Beweiser verifizieren zu lassen.

3.6.30.3.2 Anforderungen an die Methodenunterstützung

LSE30.M.1 Designverifikation (DVER)

LSE30.M.1.1 Zusammenspiel formaler Methoden
Werden mehrere formale Methoden verwendet, so sind Aussagen über deren Zusammenspiel von dem Werkzeug mathematisch exakt beweisbar.

LSE30.M.1.2 Unterstützung der formalen Spezifikationssprache
Die verwendete(n) formale(n) Spezifikationssprache(n) wird/werden unterstützt.

LSE30.M.2 Programmverifikation (PVER)

LSE30.M.2.1 Zusammenspiel formale Spezifikation und Zielsprache
Sowohl die Zielsprache als auch die formale Spezifikationssprache werden unterstützt.

3.6.30.3.3 Anforderungen an die Funktionen

LSE30.F.1 Generierung von Verifikationsbedingungen
Es ist möglich, Verifikationsbedingungen zu generieren, mit denen bewiesen wird, daß das Programm die Spezifikation erfüllt bzw. die Spezifikation eine Verfeinerung ist. Diese entstehen beim Beweis des Sicherheitsmodells, bei jeder Verfeinerung einer Spezifikation (DVER) und beim Beweis, daß ein Programm eine Spezifikation erfüllt (PVER).

LSE30.F.2 Simplifier
Es ist möglich, die erzeugten Verifikationsbedingungen gemäß einem Kalkül bzw. vorgegebenen Regeln automatisch zu beweisen bzw. zu vereinfachen.

LSE30.F.3 Kalkül

LSE30.F.3.1 Vollständigkeit
Der Kalkül unterstützt alle Operatoren der Spezifikationssprache und bei PVER zusätzlich alle verwendeten Operatoren der Zielsprache.

LSE30.F.3.2 Konsistenz
Der Kalkül ist in sich konsistent.

LSE30.F.4 Interaktiver Beweiser

LSE30.F.4.3 Kompatibilität
Alle Verifikationsbedingungen, die durch die eingesetzten Verifikationsmethoden erzeugt werden, sind mit dem Werkzeug in die Eingabesprache des interaktiven Beweisers transformierbar.

LSE30.F.4.4 Beweis der Verifikationsbedingungen
Der Beweiser unterstützt den Beweis aller Verifikationsbedingungen.

LSE30.F.4.4.1 Prädikatenlogik erster Stufe
Beweiser für Prädikatenlogik erster Stufe ist möglich.

LSE30.F.4.4.2 In die Werkzeuge eingebundene Theorien
Beweiser für die in die Werkzeuge eingebundenen Theorien (z. B. natürliche Zahlen) ist möglich.

LSE30.F.4.4.3 Behandlung der Gleichheit
Behandlung der Gleichheit ist möglich.

LSE30.F.4.4.4 Induktionsbeweiser
Induktionsbeweiser ist möglich.

LSE30.F.4.5 Heuristiken
Das Werkzeug unterstützt geeignete Beweis-Heuristiken bzw. Strategien.

LSE30.F.4.6 Regelauswahl
Es ist möglich, Hilfe zu geben, welche Regeln angewendet werden können.

LSE30.F.5 Editor

LSE30.F.5.1 Eingabe von Formeln
Es gibt nutzerfreundliche Hilfsmittel zur Eingabe der Beweisschritte.

Beispielsweise sollte ein einheitlicher, leicht handhabbarer Editor zur Verfügung stehen.

LSE30.F.5.2 Strukturierte Darstellung
Für Ausgaben des Werkzeugs ist deutlich, in welchem Kontext sie stehen. Sie sind gut lesbar, leicht verständlich und präzise. Die Beweise sind von einem Nutzer nachvollzieh-

bar. Dies bedeutet auch, daß zu den einzelnen Beweisschritten die verwendeten Regeln angegeben sind.

LSE30.F.6 Formelsammlungen

LSE30.F.6.1 Datenbank
Es gibt Datenbanken (Formelsammlungen), in denen Basis-Bausteine wie formal spezifizierte Module, elementare Datenstrukturen mit entsprechenden Theorien, Theoreme und Lemmata abgelegt sind, auf die die Komponenten des Werkzeugs zugreifen können.

LSE30.F.6.1 Verifizierte/nicht verifizierte Bausteine
Es ist möglich, verifizierte von nicht verifizierten Bausteinen zu unterscheiden.

LSE30.F.7 Protokollierung
Alle Entwicklungsschritte und interaktiv getroffenen Nutzerentscheidungen werden gespeichert, um den Entwicklungsstand nachvollziehbar darzustellen und die Fehlersuche zu erleichtern. Insbesondere ist es möglich festzuhalten, wie weit die Spezifikation bzw. das Programm verifiziert ist.

LSE30.F.8 Backtracking
Es ist möglich, in einem Entwicklungsvorgang zurückzugehen und an den entsprechenden Stellen neue Nutzerentscheidungen zuzulassen.

LSE30.F.9 Replay
Ohne Interaktion mit dem Nutzer werden gespeicherte Entwicklungsvorgänge bis zum aktuellen Stand durch das Werkzeug nachvollzogen und wiederhergestellt.

3.6.30.3.4 Sonstige Anforderungen

LSE30.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE30.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen. Die Syntax dieser Sprache muß formal definiert sein.

LSE30.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LSE30.So.4 Einheitliche Nutzerschnittstelle

LSE30.So.4.1 Die Oberfläche ist für alle Teile des Verifikationswerkzeugs einheitlich.

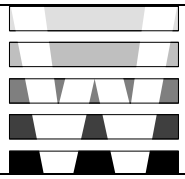
LSE30.So.4.2 Die Oberfläche des Spezifikationswerkzeugs und des Verifikationswerkzeugs sind gleich.

LSE30.So.5 Effizienz
Mit dem Werkzeug sind – gemessen am Stand der Technik – Entwicklungen in akzeptabler Zeit und mit akzeptablem Speicheraufwand durchführbar.

LSE30.So.6 Installation

LSE30.So.6.1 Verfahren
Es ist ein Verfahren zur Installation des Verifikationswerkzeugs festgelegt und mit sämtlichen Parametern dokumentiert.

LSE30.So.6.2 Konfigurationsmöglichkeiten
Alle Konfigurationsmöglichkeiten für den Werkzeug-Start durch den Nutzer und die ent-



sprechenden Parameter sind mit ihren Auswirkungen auf das Gesamtverhalten des Werkzeugs dokumentiert.

LSE30.So.6.3 Authentizität des Binärcodes
Die Verfahren zur Sicherstellung der Authentizität des Binärcodes, zur Installation und zum Start eines Werkzeugs werden beschrieben. Das Vorgehen bei der Wartung des Werkzeugs mit der entsprechenden Versionskontrolle ist dokumentiert.

LSE30.So.7 Dokumentation

LSE30.So.7.1 Nutzerdokumentation
Die Nutzerdokumentation wendet sich an einen Nutzer mit Kenntnissen zur formalen Spezifikation und beinhaltet alle zur Arbeit mit dem Werkzeug nötigen Informationen.

LSE30.So.7.2 Kalkül
Die Regelsprache bzw. der Kalkül sind formal und verbal beschrieben (sowohl Syntax als auch Semantik).

LSE30.So.7.3 Sonstige Dokumentation
Die verwendete Methodik und mathematischen Grundlagen werden präzise beschrieben.

LSE30.So.8 Zulassung beim BSI
Für Entwicklungen nach E5 oder E6 gemäß ITSEC ist das Werkzeug vom BSI zugelassen.

3.6.31 Leistungseinheit LSE31 „Analyse verdeckter Kanäle“

3.6.31.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- SE 2.5 Schnittstellen beschreiben
  („Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)
- SE 4.1-SW SW-Architektur entwerfen
  („SW-Architektur.*Lösungsvorschläge*“,
  „SW-Architektur.*Modularisierung/Datenbankentwurf*“,
  „SW-Architektur.*Schnittstellen*“ und
  „Schnittstellenübersicht.*Systeminterne Schnittstellen*“)
- SE 4.2-SW SW-interne und -externe Schnittstellen entwerfen
  („Schnittstellenbeschreibung.*Beschreibung der Schnittstellen*“)
- SE 5.1-SW SW-Komponente/-Modul/Datenbank beschreiben
  („SW-Entwurf.*SW-Komponenten-/SW-Modul-Beschreibung*“ und
  „SW-Entwurf.*Datenbankbeschreibung*“ [bei objektorientierten Datenbanken])
- SE 5.2-SW Betriebsmittel- und Zeitbedarf analysieren
  („SW-Entwurf.*Kenngrößen*“)
- SE 6.1-SW SW-Module codieren
  („SW-Modul“)
- SE 6.2-SW Datenbank realisieren
  („Datenbank“)
- SE 6.3-SW Selbstprüfung des SW-Moduls/der Datenbank durchführen
  („Implementierungsdokumente (SW-Modul)“,
  „Implementierungsdokumente (Datenbank)“,
  „SW-Modul“ und
  „Datenbank“)

Methode

- Analyse verdeckter Kanäle (AVK)

3.6.31.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur Unterstützung der „Analyse verdeckter Kanäle“ gestellt (AVK). Die Analyse verdeckter Kanäle spielt dann eine Rolle, wenn der Informationsfluß in den Sicherheitsvorgaben geregelt wird. Die Funktionalitätsklassen F-B2 und F-B3 gemäß ITSEC fordern eine solche Analyse.

Die Analyse besteht aus den folgenden Schritten:

1. Identifizieren der verdeckten Kanäle,
2. Ermitteln von Szenarien, um die Kanäle auszunutzen,
3. Abschätzen der Bandbreite der Kanäle,
4. Planung und Einsatz von Gegenmaßnahmen.

Ein Werkzeug sollte insbesondere die Schritte 1 bis 3 unterstützen, wobei die Hauptaufgabe des Werkzeugs im 1. Schritt zu sehen ist, der sich weiter unterteilen läßt in:

a) Identifikation aller Objekte, die eventuell zur Informationsübermittlung ausgenutzt werden können,
b) Identifikation aller Operationen, die diese Objekte lesen oder modifizieren,
c) Analyse der Informationsflüsse zwischen diesen Objekten bei der Ausführung der Operationen,
d) Ermitteln der Objekte, die mögliche Kandidaten für die unerlaubte Informationsübermittlung sind, bzw. Generieren von Verifikationsbedingungen, die bewiesen werden müssen, um zu zeigen, daß das System sicher ist,
e) Untersuchung, ob die gefundenen möglichen Kandidaten als verdeckte Kanäle ausgenutzt werden können.

3.6.31.3 Anforderungen

3.6.31.3.1 Anforderungen an die Schnittstellen

LSE31.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LSE31.S.2 Eingangs-Schnittstelle zu LSE23 „Subsystemmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE23 definierten logischen Gliederung des Systems zuzugreifen.

LSE31.S.3 Eingangs-Schnittstelle zu LSE24 „Moduldiagramme unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE24 definierten Modulstruktur zuzugreifen.

LSE31.S.4 Eingangs-Schnittstelle zu LSE25 „Prozeßdiagramme unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu der mit LSE25 definierten Prozeßstruktur zuzugreifen.

LSE31.S.5 Eingangs-Schnittstelle zu LSE26 „Use-Case-Modellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE26 definierten Use Cases zuzugreifen.

LSE31.S.6 Eingangs-Schnittstelle zu LSE27 „Zustandsmodellierung im objektorientierten Bereich unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE27 definierten Zuständen, Ereignissen und Aktionen zuzugreifen.

LSE31.S.7 Eingangs-Schnittstelle zu LSE28 „Interaktionsmodellierung unterstützen“
Es ist möglich, über die Objektverwaltung auf Informationen zu den mit LSE28 definierten Objekten und Interaktionen zuzugreifen.

LSE31.S.8 Eingangs-Schnittstelle zu LSE29 „Formal spezifizieren“
Es ist möglich, die erstellte formale Spezifikation von dem Werkzeug zur Analyse verdeckter Kanäle weiterverarbeiten zu lassen.

LSE31.S.9 Ausgangs-Schnittstelle zu LSE30 „Formal verifizieren“
Es ist möglich, die im Werkzeug zur Analyse verdeckter Kanäle erstellten Verifikationsbedingungen im Beweiser der „Formalen Verifikation“ zu verifizieren. Alternativ ist es möglich, daß auch das Werkzeug zur Analyse verdeckter Kanäle einen solchen Beweiser besitzt.

3.6.31.3.2 Anforderungen an die Methodenunterstützung

LSE31.M.1 Analyse verdeckter Kanäle (AVK)

LSE31.M.1.1 Unterstützung der formalen Spezifikationssprache
Die verwendete(n) formale(n) Spezifikationssprache(n) wird/werden unterstützt.

D. h. aus ihr muß die Analyse verdeckter Kanäle (Speicherkanäle) abgeleitet werden können.

LSE31.M.1.2 Unterstützung der Zielsprache
Es ist möglich, die Analyse verdeckter Kanäle (Speicherkanäle) aus einem Programm in der Zielsprache abzuleiten.

LSE31.M.1.3 Sicherheitspolitik
Die Analyse verdeckter Kanäle ist abhängig von der Sicherheitspolitik bzw. den Sicherheitsvorgaben. Das Werkzeug ist nicht auf eine bestimmte Sicherheitspolitik abgestimmt, sondern parametrisierbar.

3.6.31.3.3 Anforderungen an die Funktionen

LSE31.F.1 Identifikation von Objekten
Es ist möglich, Objekte aus der formalen Spezifikation bzw. dem Programm zu identifizieren, die als verdeckte Speicherkanäle ausgenutzt werden können.

LSE31.F.2 Identifikation von Operationen
Es ist möglich, die möglichen Operationen zu ermitteln.

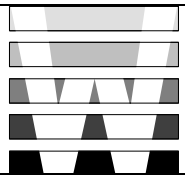
LSE31.F.3 Identifikation von Informationsflüssen
Es ist möglich, die Informationsflüsse zwischen den Objekten auf Grund der Ausführung von Operationen zu ermitteln.

LSE31.F.4 Generierung von Verifikationsbedingungen
Es ist möglich, Verifikationsbedingungen zu generieren, die eingehalten werden müssen, damit das System sicher (d. h. ohne verdeckte Kanäle) ist. Diese können mit dem Simplifier und dem Beweiser in LSE 30 „Formal verifizieren“ bewiesen werden.

LSE31.F.5 Ermitteln von Kandidaten für verdeckte Kanäle
Aus der Informationsflußanalyse und den Verifikationsbedingungen werden mögliche Kandidaten für verdeckte Kanäle ermittelt.

LSE31.F.5.1 Vollständigkeit
Es ist möglich, alle Objekte aufzulisten, über die Speicherkanäle (und eventuelle Zeitkanäle) möglich sind.

LSE31.F.5.2 Genauigkeit
Es ist möglich, möglichst nur die Objekte aufzulisten, über die verdeckte Kanäle auch wirklich möglich sind.



LSE31.F.6 Entwickeln möglicher Szenarien
Es ist möglich, Szenarien zu entwickeln, unter denen die Objekte eventuell als verdeckte Kanäle ausgenutzt werden können.

LSE31.F.7 Bestimmung der Bandbreite
Es ist möglich, bei der Bestimmung der Bandbreite der verdeckten Kanäle Unterstützung zu bieten.

LSE31.F.8 Editor

LSE31.F.8.1 Eingabe von Objekten, Operationen, Informationsflüssen
Es gibt nutzerfreundliche Hilfsmittel zur Eingabe von Objekten, Operationen und Informationsflüssen, damit das Werkzeug auch auf informelle Spezifikationen angewendet werden kann.

Beispielsweise sollte ein einheitlicher, leicht handhabbarer Editor zur Verfügung stehen.

LSE31.F.8.2 Strukturierte Darstellung
Für Ausgaben des Werkzeugs ist deutlich, in welchem Kontext sie stehen. Sie müssen gut lesbar, leicht verständlich und präzise sein. Die Beweise sind von einem Nutzer nachvollziehbar. Dies bedeutet auch, daß zu den einzelnen Beweisschritten die verwendeten Regeln angegeben sind.

3.6.31.3.4 Sonstige Anforderungen

LSE31.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LSE31.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LSE31.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LSE31.So.4 Einheitliche Nutzerschnittstelle

LSE31.So.4.1 Die Oberfläche ist für alle Teile des Werkzeugs zur Analyse verdeckter Kanäle einheitlich.

LSE31.So.4.2 Die Oberfläche ist mit der des Spezifikationswerkzeugs und des Verifikationswerkzeugs gleich.

LSE31.So.5 Effizienz

LSE31.So.5.1 Mit dem Werkzeug sind – gemessen am Stand der Technik – Entwicklungen in akzeptabler Zeit und mit akzeptablem Speicheraufwand durchführbar..

LSE31.So.6 Installation

LSE31.So.6.1 Verfahren
Es ist ein Verfahren zur Installation des Werkzeugs zur Analyse verdeckter Kanäle festgelegt und mit sämtlichen Parametern dokumentiert.

LSE31.So.6.2 Konfigurationsmöglichkeiten
Alle Konfigurationsmöglichkeiten für den Werkzeug-Start durch den Nutzer und die entsprechenden Parameter sind mit ihren Auswirkungen auf das Gesamtverhalten des Werkzeugs dokumentiert.

LSE31.So.6.3 Authentizität des Binärcodes
Die Verfahren zur Sicherstellung der Authentizität des Binärcodes, zur Installation und

zum Start eines Werkzeugs werden beschrieben . Das Vorgehen bei der Wartung des Werkzeugs mit der entsprechenden Versionskontrolle ist dokumentiert.

LSE31.So.7 Dokumentation

LSE31.So.7.1 Nutzerdokumentation
Die Nutzerdokumentation wendet sich an einen Nutzer mit Kenntnissen zur formalen Spezifikation und beinhaltet alle zur Arbeit mit dem Werkzeug nötigen Informationen.

LSE31.So.7.2 Sonstige Dokumentation
Die verwendete Methodik und mathematischen Grundlagen werden präzise beschrieben.

LSE31.So.8 Unabhängigkeit von Beweiser und Spezifikationssprachen bzw. Zielsprachen
Das Werkzeug ist unabhängig von den Beweisern, Spezifikationssprachen und Zielsprachen.

D. h. die Schnittstellen zu Beweisern, Spezifikationssprachen und Zielsprachen sollten möglichst klein gehalten werden. Notwendige Abhängigkeiten sollten sich auf einzelne Module beschränken, die gegebenenfalls ausgetauscht werden können.

3.7 V-Modell: Leistungskomplex QS

3.7.1 Leistungseinheit LQS01 „Prüfungen der Anforderungen unterstützen“

3.7.1.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.1 Prüfbarkeit feststellen
(Prüfgegenstände
„Anwenderforderungen“ und
„Technische Anforderungen“)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstände
„Anwenderforderungen“ und
„Technische Anforderungen“)

Methode

– Statische Analyse (STAT)

3.7.1.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen die Qualitätssicherung für folgende Produkte unterstützt wird:

– Anwenderforderungen
– Technische Anforderungen

Die obigen Produkte werden auf Konsistenz und Vollständigkeit überprüft.

3.7.1.3 Anforderungen

3.7.1.3.1 Anforderungen an die Schnittstellen

LQS01.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.1.3.2 Anforderungen an die Methodenunterstützung

LQS01.M.1 Statische Analyse (STAT)

LQS01.M.1.1 Formalisierte Anforderungsbeschreibungen
Es ist möglich, die Anforderungsbeschreibungen auf Einhaltung eines vorgegebenen Formalismus’ zu überprüfen.

LQS01.M.1.2 Prüfung von Konventionen

LQS01.M.1.2.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS01.M.1.2.2 Namenskonventionen
Es ist möglich, Namenskonventionen zu überprüfen.

LQS01.M.1.2.3 Beschränkungen
Es ist möglich, Komplexitätsbeschränkungen zu überprüfen.

LQS01.M.1.2.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen zu überprüfen.

LQS01.M.1.2.5 Darstellungsformen
Es ist möglich zu überprüfen, ob einmal festgelegte Darstellungsformen eingehalten werden.

LQS01.M.1.3 Anforderungszuordnung

LQS01.M.1.3.1 Zuordnungsgegenstand
Es ist möglich zu überprüfen, ob jede Anforderung einem Segment, einer SW-Einheit, einer Schnittstelle oder der Entwicklungs- und SWPÄ-Umgebung zugeordnet wurde.

LQS01.M.1.3.2 Bereits zugeordnet
Es ist möglich, bereits zugeordnete Anforderungen anzeigen zu lassen.

LQS01.M.1.3.3 Noch nicht zugeordnet
Es ist möglich, noch nicht zugeordnete Anforderungen anzeigen zu lassen.

LQS01.M.1.3.4 Herkunftsprüfung
Es ist möglich, die Herkunft aller Anforderungen zu prüfen.

Es muß z. B. möglich sein zu überprüfen, ob die Anforderung aus Anforderungen einer höheren Ebene abgeleitet ist, ob es sich um eine Nachforderung handelt, usw.

LQS01.M.1.4 Anpassung
Es ist möglich, individuell angepaßte Darstellungsformen des zu prüfenden Objektes innerhalb der Prüfung mit zu berücksichtigen.

3.7.1.3.3 Anforderungen an die Funktionen

LQS01.F.1 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

LQS01.F.2 Prozedursprache
Es ist möglich, weitere individuelle Prüfungen mit Hilfe einer Prozedursprache durchzuführen.

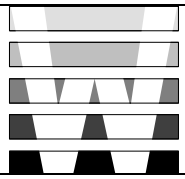
Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS01.F.3 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.1.3.4 Sonstige Anforderungen

LQS01.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.



LQS01.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS01.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS01.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS01.So.5 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.2 Leistungseinheit LQS02 „Prüfungen der Nutzerschnittstellen unterstützen“

3.7.2.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.1 Prüfbarkeit feststellen
(Prüfgegenstände
„Technische Anforderungen.*Technische Anforderungen an die Nutzerschnittstelle*“
und
„Schnittstellenbeschreibung“)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstände
„Technische Anforderungen.*Technische Anforderungen an die Nutzerschnittstelle*“
und
„Schnittstellenbeschreibung“)

Methode

– Statische Analyse (STAT)

3.7.2.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen geprüft wird, ob alle Inhalte und Nutzerelemente der Nutzerschnittstellen beschrieben worden sind. Weiterhin wird die Struktur des Nutzerdialogs geprüft.

3.7.2.3 Anforderungen

3.7.2.3.1 Anforderungen an die Schnittstellen

LQS02.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

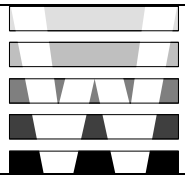
3.7.2.3.2 Anforderungen an die Methodenunterstützung

LQS02.M.1 Statische Analyse (STAT)

LQS02.M.1.1 Dialog Design Modellierung (DIAL)

LQS02.M.1.1.1 Bildschirmmaskentypen
Es ist möglich aufzulisten, welche für einen geplanten Bildschirmdialog relevanten Bildschirmmaskentypen nicht in Dialog-Diagrammen zu finden sind.

LQS02.M.1.1.2 Übergangsereignisse
Es ist möglich, die Übergänge von einem Maskentyp zum anderen aufzulisten, bei denen die Angabe des Übergangsereignisses (Auslöser) im Dialog-Diagramm fehlt.



LQS02.M.1.1.3 Prüfung von Konventionen

LQS02.M.1.1.3.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS02.M.1.1.3.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS02.M.1.1.3.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS02.M.1.1.3.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS02.M.1.1.3.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

3.7.2.3.3 Anforderungen an die Funktionen

LQS02.F.1 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

LQS02.F.2 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS02.F.3 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.2.3.4 Sonstige Anforderungen

LQS02.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS02.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS02.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS02.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS02.So.5 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.3 Leistungseinheit LQS03 „Prüfungen der Architektur unterstützen“

3.7.3.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.1 Prüfbarkeit feststellen
(Prüfgegenstand „SW-Architektur.*Modularisierung/Datenbankentwurf*“)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstand „SW-Architektur.*Modularisierung/Datenbankentwurf*“)

Methode

– Statische Analyse (STAT)

3.7.3.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge zur

– Untergliederung jeder SW-Einheit in SW-Komponenten und SW-Module,

– zur vollständigen Beschreibung der Kommunikationsbeziehungen

gestellt.

3.7.3.3 Anforderungen

3.7.3.3.1 Anforderungen an die Schnittstellen

LQS03.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.3.3.2 Anforderungen an die Methodenunterstützung

LQS03.M.1 Statische Analyse (STAT)

LQS03.M.1.1 Objektentwurfstechnik (OBJE)

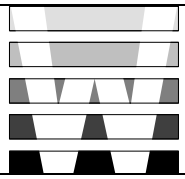
LQS03.M.1.1.1 Benutzt-Hierarchie

LQS03.M.1.1.1.1 Benutzung passiver Objekte
Es ist möglich zu prüfen, ob passive Objekte nur Operationen anderer passiver Objekte benutzen.

Es wird bei der Methode OBJE nicht empfohlen, daß passive Objekte Operationen von aktiven Objekten benutzen.

LQS03.M.1.1.1.2 Zyklen
Es ist möglich zu prüfen, ob die Benutzt-Hierarchie passiver Objekte untereinander keinen Zyklus enthält. Eventuelle Zyklen werden aufgelistet.

LQS03.M.1.1.1.3 Datenfluß
Es ist möglich zu prüfen, ob ein Datenfluß angegeben wurde, wenn ein Objekt Operationen eines anderen Objekts benutzt.



LQS03.M.1.1.1.4 Ausnahme
Es ist möglich zu prüfen, ob die Ausnahme nur dort angegeben wurde, wo ein Objekt Operationen eines anderen Objekts benutzt.

LQS03.M.1.1.2 Besteht-aus-Hierarchie

LQS03.M.1.1.2.1 Elternobjekt
Es ist möglich zu prüfen, ob ein Objekt maximal ein ihn umgebendes Objekt hat.

Das umgebende Objekt wird auch „Elternobjekt“ genannt.

LQS03.M.1.1.2.2 Selbstenthaltensein
Es ist möglich zu prüfen, ob ein Objekt sich nicht selbst enthält.

LQS03.M.1.1.3 Objekte

LQS03.M.1.1.3.1 Namensgebung
Es ist möglich zu prüfen, ob der Name jedes Objekts einzigartig innerhalb des Objektdiagramms ist. Eventuelles Mehrfachauftreten wird aufgelistet.

LQS03.M.1.1.3.2 Wurzelobjekt
Es ist möglich zu prüfen, ob das Objektdiagramm genau ein Wurzelobjekt besitzt.

Ein Wurzelobjekt hat kein übergeordnetes Objekt.

LQS03.M.1.1.4 Operationen

LQS03.M.1.1.4.1 Namensgebung
Es ist möglich zu prüfen, ob der Name jeder Operation einzigartig innerhalb eines Objekts ist. Eventuelles Mehrfachauftreten wird aufgelistet.

LQS03.M.1.1.4.2 Zuordnung
Es ist möglich zu prüfen, ob jede Operation von genau einem Objekt zur Verfügung gestellt wird.

LQS03.M.1.1.4.3 Ausnahmebehandler
Es ist möglich zu prüfen, ob ein aktives Objekt, dem eine Ausnahme angestoßen werden kann, einen Ausnahmebehandler für diese Ausnahme besitzt.

LQS03.M.1.1.5 Schnittstellen
Es ist möglich zu prüfen, ob die zur Verfügung gestellten Schnittstelle des benutzten Objekts mit der benötigten Schnittstelle des benutzenden Objekts zusammenpaßt. Objekte mit nicht zusammenpassenden Schnittstellen werden aufgelistet.

LQS03.M.2 Structured Design (STRD)

LQS03.M.2.1 Verwendungsprüfung
Es ist möglich, SW-Komponenten und SW-Module, die in keiner Architektur enthalten sind, auflisten zu lassen.

LQS03.M.2.2 Hierarchieprüfung
Es ist möglich, die Architektur auf ihren „ruft-auf“-Charakter prüfen zu lassen.

LQS03.M.2.3 Konstruktprüfung
Es ist möglich, die Einhaltung der Zeitreihenfolgekonstrukte zu prüfen.

LQS03.M.2.4 Prüfung von Konventionen

LQS03.M.2.4.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS03.M.2.4.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS03.M.2.4.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS03.M.2.4.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzurüfen.

LQS03.M.2.4.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

3.7.3.3.3 Anforderungen an die Funktionen

LQS03.F.1 Prüfung auf Vollständigkeit

LQS03.F.1.1 Kopplung
Es ist möglich, die Kopplungen auf vollständige Beschreibung zu prüfen.

LQS03.F.1.2 Aufruf
Es ist möglich, die Aufrufe auf vollständige Beschreibung zu prüfen.

LQS03.F.1.3 Beschreibung
Es ist möglich, Objekte auflisten zu lassen, zu denen keine Beschreibung existiert.

LQS03.F.2 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

LQS03.F.3 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS03.F.4 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.3.3.4 Sonstige Anforderungen

LQS03.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

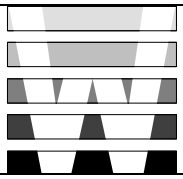
LQS03.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS03.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS03.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.



LQS03.So.5 Invarianz

Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.4 Leistungseinheit LQS04 „Prüfungen der Prozeßmodelle unterstützen“

3.7.4.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.1 Prüfbarkeit feststellen
(Prüfgegenstand
„SW-Architektur.*Modularisierung*“)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstand
„SW-Architektur.*Modularisierung*“)

Methode

– Statische Analyse (STAT)

3.7.4.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen das mit Hilfe der Leistungseinheit LSE04 „Prozeßmodellierung unterstützen“ erstellte Prozeßmodell statisch überprüft werden kann. Im Rahmen dieser Prüfung wird das Prozeßmodell auf

– syntaktische Korrektheit,
– statische Eigenschaften wie Konsistenz und Vollständigkeit und
– Einhaltung von Standards und Richtlinien

überprüft.

3.7.4.3 Anforderungen

3.7.4.3.1 Anforderungen an die Schnittstellen

LQS04.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

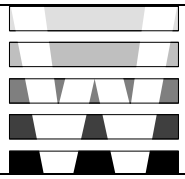
3.7.4.3.2 Anforderungen an die Methodenunterstützung

LQS04.M.1 Statische Analyse (STAT)

LQS04.M.1.1 Prüfung der formalen und syntaktischen Korrektheit

LQS04.M.1.1.1 Anordnungsregeln
Die Einhaltung der Anordnungsregeln wird geprüft.

LQS04.M.1.1.2 Vollständigkeit
Die Vollständigkeit der Angaben zu den einzelnen Objekten wird geprüft.



LQS04.M.1.2 Prüfung der Konsistenz
Die Konsistenz zwischen PZIM und ZUST wird geprüft.

Es wird u. a. geprüft, ob jeder Prozeß der Prozeßinteraktionsmodellierung PZIM durch die Zustandsübergangsmodellierung ZUST einzeln spezifiziert wird und ob eingehende Signalpfeile eines Prozesses in PZIM als Eingabe in ZUST und ausgehende Signalpfeile eines Prozesses in PZIM als Ausgabe in ZUST auftreten.

LQS04.M.1.3 Prüfung von Konventionen

LQS04.M.1.3.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS04.M.1.3.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS04.M.1.3.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS04.M.1.3.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS04.M.1.3.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

LQS04.M.1.4 Schnittstellenkonsistenz bei Hierarchisierung
Die Schnittstellen von einer Ebene zur nächsten werden automatisch auf Konsistenz geprüft.

LQS04.M.1.5 Schnittstellenkonsistenz bei Teilmodellbildung
Die Schnittstellen von einem Teilmodell zum anschließenden werden automatisch auf Konsistenz geprüft.

LQS04.M.1.6 Vollständigkeit der Beschreibung
Es ist möglich, sich die Objekte auflisten zu lassen, zu denen keine Beschreibungen existieren.

LQS04.M.1.7 Anpassung
Individuell angepaßte Darstellungsformen des zu prüfenden Objektes werden innerhalb der Prüfung mit berücksichtigt.

3.7.4.3.3 Anforderungen an die Funktionen

LQS04.F.1 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS04.F.2 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

LQS04.F.3 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS04.F.4 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.4.3.4 Sonstige Anforderungen

LQS04.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS04.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS04.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS04.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS04.So.5 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.5 Leistungseinheit LQS05 „Prüfungen der Funktionsstrukturen unterstützen“

3.7.5.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.1 Prüfbarkeit feststellen
(Prüfgegenstände
„Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“,
„Anwenderforderungen.*Grobe Systembeschreibung*“,
„Anwenderforderungen.*Beschreibung der Funktionalität*“,
„Systemarchitektur.*Lösungsvorschläge*“,
„Systemarchitektur.*Struktur des Systems*“ und
„Technische Anforderungen.*Gesamtfunktion*“)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstände
„Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“,
„Anwenderforderungen.*Grobe Systembeschreibung*“,
„Anwenderforderungen.*Beschreibung der Funktionalität*“,
„Systemarchitektur.*Lösungsvorschläge*“,
„Systemarchitektur.*Struktur des Systems*“ und
„Technische Anforderungen.*Gesamtfunktion*“)

Methode

– Statische Analyse (STAT)

3.7.5.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen u. a. überprüft werden kann, ob alle Funktionen festgelegten Konventionen (Namenskonventionen, Projektstandards, usw.) genügen und hierarchisch gegliedert sind.

3.7.5.3 Anforderungen

3.7.5.3.1 Anforderungen an die Schnittstellen

LQS05.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.5.3.2 Anforderungen an die Methodenunterstützung

LQS05.M.1 Statische Analyse (STAT)

LQS05.M.1.1 Funktionale Dekomposition (FKTD)

LQS05.M.1.1.1 Isolierte Ojekte
Es ist möglich, die Objekte aufzulisten, die in keine Funktionshierarchie aufgenommen wurden.

LQS05.M.1.1.2 Hierarchie
Es ist möglich, die Objekte aufzulisten, denen mehr als ein hierarchisch übergeordnetes Objekt zugerechnet wurde.

LQS05.M.1.1.3 Prüfung von Konventionen

LQS05.M.1.1.3.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS05.M.1.1.3.2 Namenskonventionen
Namenskonventionen werden abgeprüft.

LQS05.M.1.1.3.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS05.M.1.1.3.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS05.M.1.1.3.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

LQS05.M.1.1.4 Vollständigkeit der Beschreibung
Es ist möglich, sich die Objekte auflisten zu lassen, zu denen keine Beschreibungen existieren.

LQS05.M.1.1.5 Anpassung
Es ist möglich, individuell angepaßte Darstellungsformen des zu prüfenden Objektes innerhalb der Prüfung mit zu berücksichtigen.

3.7.5.3.3 Anforderungen an die Funktionen

LQS05.F.1 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

LQS05.F.2 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS05.F.3 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

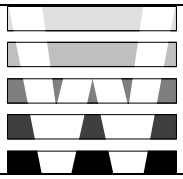
Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.5.3.4 Sonstige Anforderungen

LQS05.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS05.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS05.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.



Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS05.So.4 Endesignal

Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS05.So.5 Invarianz

Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.6 Leistungseinheit LQS06 „Prüfungen der Spezifikationen von Funktionen unterstützen“

3.7.6.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.1 Prüfbarkeit feststellen
(Prüfgegenstände
„Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“,
„Anwenderforderungen.*Grobe Systembeschreibung*“,
„Anwenderforderungen.*Beschreibung der Funktionalität*“ und
„Technische Anforderungen.*Gesamtfunktion*“)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstände
„Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“,
„Anwenderforderungen.*Grobe Systembeschreibung*“,
„Anwenderforderungen.*Beschreibung der Funktionalität*“ und
„Technische Anforderungen.*Gesamtfunktion*“)

Methode

– Statische Analyse (STAT)

3.7.6.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen u. a. überprüft werden kann, ob die Ablauflogik mit Konstrukttypen aufgebaut wurde, die dem Prinzip der Strukturierten Programmierung entsprechen.

3.7.6.3 Anforderungen

3.7.6.3.1 Anforderungen an die Schnittstellen

LQS06.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

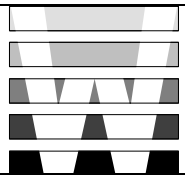
3.7.6.3.2 Anforderungen an die Methodenunterstützung

LQS06.M.1 Statische Analyse (STAT)

LQS06.M.1.1 Prüfung von Konventionen

LQS06.M.1.1.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS06.M.1.1.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.



LQS06.M.1.1.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS06.M.1.1.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS06.M.1.1.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

LQS06.M.1.2 Struktogramme (Aktionsdiagramme)

LQS06.M.1.2.1 Fehlende Struktogramme (Aktionsdiagramme)
Es ist möglich, die Funktionen aufzulisten, die nicht mit Hilfe von Struktogrammen (Aktionsdiagrammen) aufgebaut sind.

LQS06.M.1.2.2 Syntaxfehler in Struktogrammen (Aktionsdiagrammen)
Es ist möglich, innerhalb eines Struktogrammes (Aktionsdiagrammes) Verletzungen der Syntaxregeln anzeigen zu lassen.

LQS06.M.1.3 Fehlende Objekte

LQS06.M.1.3.1 Fehlender Titel
Es ist möglich, das Fehlen des Titelblocks festzustellen.

LQS06.M.1.3.2 Unvollständige Konstrukte
Es ist möglich, unvollständige Konstrukte festzustellen.

LQS06.M.1.4 Vollständigkeit der Beschreibung
Es ist möglich, sich die Objekte auflisten zu lassen, zu denen keine Beschreibungen existieren.

LQS06.M.1.5 Anpassung
Individuell angepaßte Darstellungsformen des zu prüfenden Objektes werden innerhalb der Prüfung mit berücksichtigt.

3.7.6.3.3 Anforderungen an die Funktionen

LQS06.F.2 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

LQS06.F.3 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS06.F.4 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.6.3.4 Sonstige Anforderungen

LQS06.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS06.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS06.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS06.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS06.So.5 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.7 Leistungseinheit LQS07 „Prüfungen der Informationsflußmodelle unterstützen“

3.7.7.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.1 Prüfbarkeit feststellen
(Prüfgegenstände
„Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“,
„Anwenderforderungen.*Beschreibung der Funktionalität*“,
„Schnittstellenübersicht“,
„Technische Anforderungen.*Gesamtfunktion*“,
„Technische Anforderungen.*Technische Anforderungen an die Schnittstellen*“ und
„Systemarchitektur.*Erläuterung der Zusammenarbeit der technischen Elemente*“)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstände
„Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“,
„Anwenderforderungen.*Grobe Systembeschreibung*“,
„Anwenderforderungen.*Beschreibung der Funktionalität*“,
„Schnittstellenübersicht“,
„Technische Anforderungen.*Gesamtfunktion*“,
„Technische Anforderungen.*Technische Anforderungen an die Schnittstellen*“ und
„Systemarchitektur.*Erläuterung der Zusammenarbeit der technischen Elemente*“)

Methode

– Statische Analyse (STAT)

3.7.7.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen die Prüfung der Datenfluß- und Kontrollflußmodelle unterstützt wird.

3.7.7.3 Anforderungen

3.7.7.3.1 Anforderungen an die Schnittstellen

LQS07.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.7.3.2 Anforderungen an die Methodenunterstützung

LQS07.M.1 Statische Analyse (STAT)

LQS07.M.1.1 Datenfluß-Modellierung (DFM)

LQS07.M.1.1.1 Konsistenzprüfung

LQS07.M.1.1.1.1 Isolierte Prozesse
Es ist möglich, Prozesse aus Datenfluß-Diagrammen auflisten zu lassen, die weder Empfänger noch Sender von Datenflüssen sind.

LQS07.M.1.1.1.2 Isolierte Terminatoren
Es ist möglich, Terminatoren aus Datenfluß-Diagrammen auflisten zu lassen, die weder Empfänger noch Sender von Datenflüssen sind.

LQS07.M.1.1.1.3 Isolierte Datenspeicher
Es ist möglich, Datenspeicher auflisten zu lassen, die weder Empfänger noch Sender von Datenflüssen sind.

LQS07.M.1.1.1.4 Nicht lückenlose Datenflüsse
Es ist möglich, Datenflüsse auflisten zu lassen, die nicht über alle Hierarchieebenen lückenlos definiert sind.

LQS07.M.1.1.1.5 Schnittstellenkonsistenz bei Hierarchiesierung
Die Schnittstellen von einer Ebene zur nächsten werden automatisch auf Konsistenz geprüft.

LQS07.M.1.1.2 Verwendungsprüfung

LQS07.M.1.1.2.1 Nicht verwendete Prozesse
Es ist möglich, Prozesse, die in keinem Datenfluß-Diagramm enthalten sind, auflisten zu lassen.

LQS07.M.1.1.2.2 Nicht verwendete Terminatoren
Es ist möglich, Terminatoren auflisten zu lassen, die in keinem Datenfluß-Diagramm enthalten sind.

LQS07.M.1.1.2.3 Nicht verwendete Datenspeicher
Es ist möglich, Datenspeicher auflisten zu lassen, die in keinem Datenfluß-Diagramm enthalten sind.

LQS07.M.1.1.2.4 Nicht verwendete Datenflüsse
Es ist möglich, Datenflüsse auflisten zu lassen, die in keinem Datenfluß-Diagramm enthalten sind.

LQS07.M.1.1.3 Prüfung von Konventionen

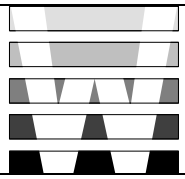
LQS07.M.1.1.3.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS07.M.1.1.3.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS07.M.1.1.3.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS07.M.1.1.3.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS07.M.1.1.3.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.



LQS07.M.1.1.4 Vollständigkeit der Beschreibung
Es ist möglich, die Objekte aufzulisten, zu denen es keine Beschreibung gibt.

LQS07.M.1.1.5 Anpassung
Es ist möglich, individuell angepaßte Darstellungsformen des zu prüfenden Objektes innerhalb der Prüfung zu berücksichtigen.

LQS07.M.1.2 Kontrollfluß-Modellierung (KFM)

LQS07.M.1.2.1 Konsistenzprüfung

LQS07.M.1.2.1.1 Isolierte Prozesse
Es ist möglich, Prozesse aus Kontrollfluß-Diagrammen auflisten zu lassen, die weder Empfänger noch Sender von Kontrollflüssen sind.

LQS07.M.1.2.1.2 Isolierte Terminatoren
Es ist möglich, Terminatoren aus Kontrollfluß-Diagrammen auflisten zu lassen, die weder Sender noch Empfänger von Kontrollflüssen sind.

LQS07.M.1.2.1.3 Isolierte Kontrollspeicher
Es ist möglich, Kontrollspeicher auflisten zu lassen, die weder Empfänger noch Sender von Kontrollflüssen sind.

LQS07.M.1.2.1.4 Nicht lückenlose Kontrollflüsse
Es ist möglich, die Kontrollflüsse auflisten zu lassen, die nicht über alle Hierarchieebenen lückenlos definiert sind.

LQS07.M.1.2.1.5 Schnittstellenkonsistenz bei Hierarchisierung
Die Schnittstelle von einer Ebene zur nächsten werden automatisch auf Konsistenz geprüft.

LQS07.M.1.2.2 Verwendungsprüfung

LQS07.M.1.2.2.1 Nicht verwendete Prozesse
Es ist möglich, Prozesse, die in keinem Kontrollfluß-Diagramm enthalten sind, auflisten zu lassen.

LQS07.M.1.2.2.2 Nicht verwendete Terminatoren
Es ist möglich, Terminatoren auflisten zu lassen, die in keinem Kontrollfluß-Diagramm enthalten sind.

LQS07.M.1.2.2.3 Nicht verwendete Kontrollspeicher
Es ist möglich, Kontrollspeicher auflisten zu lassen, die in keinem Kontrollfluß-Diagramm enthalten sind.

LQS07.M.1.2.2.4 Nicht verwendete Kontrollflüsse
Es ist möglich, Kontrollflüsse auflisten zu lassen, die in keinem Kontrollfluß-Diagramm enthalten sind.

LQS07.M.1.2.3 Prüfung von Konventionen

LQS07.M.1.2.3.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS07.M.1.2.3.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS07.M.1.2.3.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS07.M.1.2.3.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS07.M.1.2.3.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

LQS07.M.1.2.4 Vollständigkeit der Beschreibung
Es ist möglich, die Objekte aufzulisten, zu denen es keine Beschreibungen gibt.

LQS07.M.1.2.5 Anpassung
Es ist möglich, individuell angepaßte Darstellungsformen des zu prüfenden Objektes innerhalb der Prüfung zu berücksichtigen.

3.7.7.3.3 Anforderungen an die Funktionen

LQS07.F.1 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

LQS07.F.2 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS07.F.3 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.7.3.4 Sonstige Anforderungen

LQS07.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS07.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS07.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS07.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS07.So.5 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.8 Leistungseinheit LQS08 „Prüfungen der Ablaufmodelle unterstützen“

3.7.8.1 Zuordnung zum V-Modell

V-Modell-Aktivität (-Produkt)

– QS 4.1 Prüfbarkeit feststellen
(Prüfgegenstände
„Anwenderforderungen.*Beschreibung der Funktionalität*“,
„Systemarchitektur.*Erläuterung der Zusammenarbeit der technischen Elemente*“ und
„Technische Anforderungen.*Gesamtfunktion*“)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstände
„Anwenderforderungen.*Beschreibung der Funktionalität*“,
„Systemarchitektur.*Erläuterung der Zusammenarbeit der technischen Elemente*“ und
„Technische Anforderungen.*Gesamtfunktion*“)

Methode

– Statische Analyse (STAT)

3.7.8.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen u. a. geprüft wird, ob alle Ereignisse, Zustände, Bedingungen und Funktionen in Ablaufmodellen enthalten sind und ob ihre Zuordnung richtig vorgenommen wurde.

3.7.8.3 Anforderungen

3.7.8.3.1 Anforderungen an die Schnittstellen

LQS08.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.8.3.2 Anforderungen an die Methodenunterstützung

LQS08.M.1 Statische Analyse (STAT)

LQS08.M.1.1 Funktionsnetz-Modellierung (FNET)

LQS08.M.1.1.1 Prüfung von Konventionen

LQS08.M.1.1.1.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS08.M.1.1.1.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS08.M.1.1.1.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS08.M.1.1.1.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS08.M.1.1.1.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

LQS08.M.1.1.2 Schnittstellenkonsistenz bei Hierarchisierung
Die Schnittstellen von einer Ebene zur nächsten werden automatisch auf Konsistenz geprüft.

LQS08.M.1.1.3 Schnittstellenkonsistenz bei Teilmodellbildung
Die Schnittstellen von einem Teilmodell zum anschließenden werden automatisch auf Konsistenz geprüft.

LQS08.M.1.1.4 Konsistenzprüfungen

LQS08.M.1.1.4.1 Vollständigkeit des Funktionsbaumes
Es ist möglich, sich alle Funktionen auflisten zu lassen, die im Funktionsnetz verwendet, nicht aber im statischen Funktionsbaum definiert wurden.

LQS08.M.1.1.4.2 Nicht verwendete Elementarfunktionen
Es ist möglich, sich alle Elementarfunktionen auflisten zu lassen, die in keinem Funktionsnetz verwendet werden.

LQS08.M.1.1.4.3 Bedingungen für Elementarfunktionen
Es ist möglich, sich alle Elementarfunktionen auflisten zu lassen, für die keine Bedingung definiert wurde.

LQS08.M.1.1.4.4 Anfangsbedingung
Es ist möglich, sich alle Abläufe auflisten zu lassen, die nicht mit einer Daten- oder Zeitbedingung beginnen.

LQS08.M.1.1.4.5 Endebedingung
Es ist möglich, sich alle Abläufe auflisten zu lassen, die nicht mit einer Datenbedingung enden.

LQS08.M.1.1.5 Vollständigkeit der Beschreibung
Es ist möglich, sich die Objekte auflisten zu lassen, zu denen keine Beschreibungen existieren.

Dies sind u. a. Funktionen, Bedingungen, Auslöser und Verteiler.

LQS08.M.1.1.6 Anpassung
Individuell angepaßte Darstellungsformen des zu prüfenden Objektes werden innerhalb der Prüfung mit berücksichtigt.

LQS08.M.1.2 Systemverhaltensmodelle (SVM)

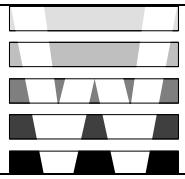
LQS08.M.1.2.1 Petri-Netze

LQS08.M.1.2.1.1 Kapazität
Es ist möglich zu prüfen, ob in einem Netz mit einer Anfangsmarkierung in allen folgenden Markierungen die Kapazität der Kanäle nicht überschritten wird.

LQS08.M.1.2.1.2 Markenerhaltung
Es ist möglich zu prüfen, ob in einem Netz mit einer Anfangsmarkierung in allen folgenden Markierungen die Anzahl der Marken konstant bleibt.

LQS08.M.1.2.1.3 Aktivierbarkeit
Es ist möglich zu prüfen, ob in einem Netz mit einer Anfangsmarkierung alle Instanzen aktivierbar sind.

LQS08.M.1.2.1.4 Verklemmungsfreiheit
Es ist möglich zu prüfen, ob in einem Netz mit einer gegebenen Anfangsmarkie-



rung in jeder Folgemarkierung mindestens eine Instanz existiert, die schalten kann.

LQS08.M.1.2.1.5 Erreichbarkeit
Es ist möglich zu prüfen, ob in einem Netz mit einer gegebenen Anfangsmarkierung eine gegebene Folgemarkierung erreicht werden kann.

LQS08.M.1.2.1.6 Prüfung von Konventionen

LQS08.M.1.2.1.6.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS08.M.1.2.1.6.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS08.M.1.2.1.6.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS08.M.1.2.1.6.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS08.M.1.2.1.6.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

LQS08.M.1.2.2 Statecharts

LQS08.M.1.2.2.1 Erreichbarkeit
Es ist möglich zu prüfen, ob in einem Statechart mit einem gegebenen Anfangszustand ein gegebener Folgezustand erreicht werden kann.

LQS08.M.1.2.2.2 Ereignisse
Es ist möglich, alle Transitionen aufzulisten, denen keine Ereignisse zugeordnet wurden.

LQS08.M.1.2.2.3 Bedingungen
Es ist möglich, alle Transitionen aufzulisten, denen keine Bedingungen zugeordnet wurden.

LQS08.M.1.2.2.4 Aktionen
Es ist möglich, alle Zustände auflisten zu lassen, denen keine Aktionen zugeordnet wurden.

LQS08.M.1.2.2.5 Aktivitäten
Es ist möglich, alle Zustände aufzulisten, denen keine Aktivitäten zugeordnet wurden.

LQS08.M.1.2.2.6 Prüfung von Konventionen

LQS08.M.1.2.2.6.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS08.M.1.2.2.6.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS08.M.1.2.2.6.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS08.M.1.2.2.6.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS08.M.1.2.2.6.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

LQS08.M.1.2.3 Specification and Description Language (SDL)

LQS08.M.1.2.3.1 Prüfung von Konventionen

LQS08.M.1.2.3.1.1 Anzeige von Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen des Werkzeugs anzuzeigen.

LQS08.M.1.2.3.1.2 Anzeige von Grenzen
Es ist möglich, die aktuelle Ausschöpfung der Größen- und Komplexitätsgrenzen beim Arbeiten mit dem Werkzeug anzuzeigen.

LQS08.M.1.2.3.1.3 Überwachen von Beschränkungen
Es ist möglich, die Überschreitung der Größen- und Komplexitätsbeschränkungen automatisch zu überwachen und im Überschreitungsfall eindeutig anzuzeigen.

LQS08.M.1.2.3.2 Schnittstellenkonsistenz bei hierarchischer Verfeinerung

LQS08.M.1.2.3.2.1 Prüfung von Kanal-Block-Schnittstellen
Bei Kanalverfeinerung werden die Kanal-Block-Schnittstellen automatisch auf Konsistenz geprüft.

LQS08.M.1.2.3.2.2 Prüfung von Schnittstellen
Bei gleichzeitigem Vorliegen von Teilbäumen und einem gesamten Baumdiagramm für Blöcke oder Prozesse werden die Schnittstellen automatisch auf Konsistenz geprüft.

LQS08.M.1.2.3.2.3 Prüfung von Hierarchieebenen
Es ist möglich, bei Prozeß-Block-Zuordnungen die Hierarchieebenen auf Konsistenz zu prüfen.

LQS08.M.1.2.3.2.4 Prüfung auf vorhandenen Prozeß
Es ist möglich, bei Prozeß-Block-Zuordnungen zu prüfen, ob allen terminalen Blöcken ein Prozeß zugeordnet ist.

LQS08.M.1.2.3.3 Konsistenzprüfungen

LQS08.M.1.2.3.3.1 Syntaxprüfung
Die Syntaxprüfung einer gesamten Systembeschreibung erstreckt sich über vorhandene Konnektoren hinweg.

LQS08.M.1.2.3.3.2 Prüfung der Sichtbarkeit
Die Sichtbarkeit von Identifikatoren für Typen, Signale, Variablen, formale Parameter, Prozeduren und Makros wird bei jedem Vorkommen überprüft.

LQS08.M.1.2.3.3.3 Prüfung der eingeschränkten Anwendbarkeit
Eine eingeschränkte Anwendbarkeit von Identifikatoren für Typen, Signale, Variablen, formale Parameter, Prozeduren und Makros wird bei gegebener Sichtbarkeit bei jedem Vorkommen überprüft.

LQS08.M.1.2.3.4 Vollständigkeit der Beschreibung

LQS08.M.1.2.3.4.1 Blöcke und Prozesse
Es ist möglich, Hinweise auf nicht beschriebene Blöcke und Prozesse anzuzeigen.

LQS08.M.1.2.3.4.2 Signale
Es ist möglich, Hinweise auf nicht benutzte Signale anzuzeigen.

LQS08.M.1.2.3.4.3 Input
Es ist möglich, Hinweise auf Zustände ohne Input-Signalangabe anzuzeigen.

LQS08.M.1.2.3.4.4 Start-/Stopsymbole
Es ist möglich, Hinweise auf fehlende Start- oder Stopsymbole anzuzeigen.

3.7.8.3.3 Anforderungen an die Funktionen

LQS08.F.1 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

LQS08.F.2 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS08.F.3 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.8.3.4 Sonstige Anforderungen

LQS08.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS08.So.2 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS08.So.3 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS08.So.4 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.9 Leistungseinheit LQS09 „Prüfungen der Informationsstrukturen unterstützen“

3.7.9.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.1 Prüfbarkeit feststellen
(Prüfgegenstände
„Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“,
„Anwenderforderungen.*Beschreibung der Funktionalität*“,
„Systemarchitektur.*Anforderungszuordnung*“,
„Technische Anforderungen.*Gesamtfunktion*“,
„Technische Anforderungen.*Technische Anforderungen an die Schnittstellen*“,
„SW-Entwurf.*Datenbankbeschreibung*“ und
„Datenkatalog“)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstände
„Anwenderforderungen.*Ist-Aufnahme und Ist-Analyse*“,
„Anwenderforderungen.*Beschreibung der Funktionalität*“,
„Systemarchitektur.*Anforderungszuordnung*“,
„Technische Anforderungen.*Gesamtfunktion*“,
„Technische Anforderungen.*Technische Anforderungen an die Schnittstellen*“,
„SW-Entwurf.*Datenbankbeschreibung*“ und
„Datenkatalog“)

Methode

– Statische Analyse (STAT)

3.7.9.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen Prüfungen von Informationsstrukturen unterstützt werden.

3.7.9.3 Anforderungen

3.7.9.3.1 Anforderungen an die Schnittstellen

LQS09.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.9.3.2 Anforderungen an die Methodenunterstützung

LQS09.M.1 Statische Analyse (STAT)

LQS09.M.1.1 E/R-Modellierung (ER)

LQS09.M.1.1.1 Konsistenzprüfungen

LQS09.M.1.1.1.1 Entitätstypen und Funktionen
Es ist möglich, die Entitätstypen auflisten zu lassen, die in den E/R-Modellen der ein- und ausgehenden Datenflüsse eines Prozesses vorkommen, nicht jedoch im E/R-Modell des Prozesses.

LQS09.M.1.1.1.2 Attribute und Funktionen
Es ist möglich, die Attribute auflisten zu lassen, die in E/R-Modellen von ein- und ausgehenden Datenflüssen eines Prozesses vorkommen, nicht jedoch im E/R-Modell des Prozesses.

LQS09.M.1.1.1.3 Entitätstypen und Datenspeicher
Es ist möglich, die Entitätstypen auflisten zu lassen, die in den E/R-Modellen der ein- und ausgehenden Datenflüsse eines Datenspeichers vorkommen, nicht jedoch im E/R-Modell des Datenspeichers.

LQS09.M.1.1.1.4 Attribute und Datenspeicher
Es ist möglich, die Attribute auflisten zu lassen, die in E/R-Modellen von ein- und ausgehenden Datenflüssen eines Datenspeichers vorkommen, nicht jedoch im E/R-Modell eines Datenspeichers.

LQS09.M.1.1.1.5 Entitätstypen mit gleichen Primärschlüsseln
Es ist möglich, Entitätstypen mit gleichem Primärschlüssel auflisten zu lassen.

Alle Attribute, die von einem Schlüssel abhängen, sollten in einer Entität zusammengefaßt werden.

LQS09.M.1.1.1.6 Attribute gleichen Namens
Es ist möglich, Attribute gleichen Namens in verschiedenen Entitätstypen auflisten zu lassen.

Attribute gleichen Namens können auf redundant definierte Attribute hinweisen.

LQS09.M.1.1.1.7 Aggregationen und Spezialisierungen

LQS09.M.1.1.1.7.1 Gleiche Attribute bei Entitätssubtypen und ihrem Supertyp
Es ist möglich, Entitätssubtypen, deren Attribute sich nicht von ihrem Supertyp unterscheiden, auflisten zu lassen.

Der Entitätssubtyp und sein Supertyp sind im Falle gleicher Attribute identisch.

LQS09.M.1.1.1.7.2 Gleiche Attribute bei mehreren Entitätssubtypen
Es ist möglich, einen Hinweis auf gleiche Attribute bei mehreren Entitätssubtypen zu erhalten.

Gleiche Attribute bei mehreren Entitätssubtypen können ein Indiz dafür sein, daß das Attribut beim Supertyp angegeben werden sollte.

LQS09.M.1.1.1.7.3 Gleiche Beziehungen bei mehreren Entitätssubtypen
Es ist möglich, einen Hinweis auf gleiche Beziehungen bei mehreren Entitätssubtypen zu erhalten.

Gleiche Beziehungen bei mehreren Entitätssubtypen können ein Indiz dafür sein, daß die Beziehung beim Supertyp angegeben werden sollte.

LQS09.M.1.1.1.7.4 Einzelne Entitätssubtypen
Es ist möglich, Entitätstyphierarchien, die nur aus einem Entitätssubtyp bestehen, auflisten zu lassen.

Eine Entitätstyphierarchie, die nur aus einem Entitätssubtyp besteht, ist möglicherweise unvollständig oder unkorrekt.

LQS09.M.1.1.1.8 Prüfung auf Redundanz
Es ist möglich, mehrere 1:N-Beziehungen zwischen zwei Entitätstypen auflisten zu lassen.

Mehrere 1:N-Beziehungen zwischen zwei Entitätstypen können ein Indiz dafür sein, daß eine Beziehung redundant ist.

LQS09.M.1.1.1.9 Prüfung auf 1:1-Beziehungen
Es ist möglich, 1:1-Beziehungen zwischen zwei Entitätstypen auflisten zu lassen.

1:1-Beziehungen zwischen zwei Entitätstypen können ein Indiz dafür sein, daß die beiden Entitätstypen zusammenzufassen sind.

LQS09.M.1.1.1.10 Isolierte Entitätstypen

LQS09.M.1.1.1.10.1 Gesamtmodell
Es ist möglich, Entitätstypen, die zwar im Gesamtmodell, jedoch in keinem Teildatenmodell vorkommen, auflisten zu lassen.

LQS09.M.1.1.1.10.2 Teilmodell
Es ist möglich, Entitätstypen, denen keine Beziehungen zugeordnet sind, auflisten zu lassen.

LQS09.M.1.1.2 Prüfung von Konventionen

LQS09.M.1.1.2.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS09.M.1.1.2.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS09.M.1.1.2.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS09.M.1.1.2.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS09.M.1.1.2.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

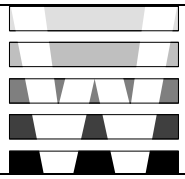
LQS09.M.1.1.3 Prüfungen auf Vollständigkeit

LQS09.M.1.1.3.1 Fehlender Datentyp
Es ist möglich, die Attribute, denen kein Datentyp zugeordnet ist, auflisten zu lassen.

LQS09.M.1.1.3.2 Fehlender Wertebereich
Es ist möglich, die Attribute, denen kein Wertebereich zugeordnet ist, auflisten zu lassen.

LQS09.M.1.1.3.3 Fehlende Attribute
Es ist möglich, die Entitätstypen, denen keine Attribute zugeordnet sind, auflisten zu lassen.

LQS09.M.1.1.3.4 Fehlende Schlüssel
Es ist möglich, die Entitätstypen, denen kein Schlüssel zugeordnet ist, auflisten zu lassen.



LQS09.M.1.1.4 Vollständigkeit der Beschreibung
Es ist möglich, sich die Objekte auflisten zu lassen, zu denen keine Beschreibungen existieren.

LQS09.M.1.1.5 Anpassung
Individuell angepaßte Darstellungsformen des zu prüfenden Objektes werden innerhalb der Prüfung mit berücksichtigt.

LQS09.M.1.2 Entity Life History (ELH)

LQS09.M.1.2.1 Prüfungen auf Vollständigkeit

LQS09.M.1.2.1.1 Fehlende Zustandsdiagramme
Es ist möglich, alle Entitätstypen aufzulisten, denen kein Zustandsdiagramm zugeordnet ist.

LQS09.M.1.2.1.2 Fehlende zustandsändernde Ereignisse
Es ist möglich, alle Zustandsänderungen aufzulisten, denen kein zustandsänderndes Ereignis zugeordnet ist.

LQS09.M.1.2.1.3 Fehlende Integritätsbedingungen
Es ist möglich, alle Entitätstypen und die dazugehörigen Zustandsänderungen aufzulisten, zu denen keine Integritätsbedingungen angegeben sind.

LQS09.M.1.2.2 Prüfung von Konventionen

LQS09.M.1.2.2.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS09.M.1.2.2.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS09.M.1.2.2.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS09.M.1.2.2.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS09.M.1.2.2.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

LQS09.M.1.2.3 Vollständigkeit der Beschreibung
Es ist möglich, die Objekte aufzulisten, zu denen es keine Beschreibung gibt.

LQS09.M.1.2.4 Anpassung
Es ist möglich, individuell angepaßte Darstellungsformen des zu prüfenden Objektes innerhalb der Prüfung zu berücksichtigen.

LQS09.M.1.3 Normalisierung (NORM)

LQS09.M.1.3.1 Konsistenzprüfungen

LQS09.M.1.3.1.1 Erste Normalform
Es ist möglich, die Nicht-Schlüsselattribute aufzulisten, die nicht vom Schlüssel funktional abhängig sind.

LQS09.M.1.3.1.2 Zweite Normalform
Es ist möglich, die Nicht-Schlüsselattribute aufzulisten, die von einem Teilschlüssel funktional abhängig sind.

LQS09.M.1.3.1.3 Dritte Normalform
Es ist möglich, die Nicht-Schlüsselattribute aufzulisten, die voneinander transitiv abhängig sind.

LQS09.M.1.3.2 Prüfung von Konventionen

LQS09.M.1.3.2.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS09.M.1.3.2.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS09.M.1.3.2.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS09.M.1.3.2.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS09.M.1.3.2.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

3.7.9.3.3 Anforderungen an die Funktionen

LQS09.F.1 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

LQS09.F.2 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS09.F.3 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.9.3.4 Sonstige Anforderungen

LQS09.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

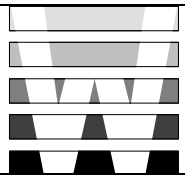
LQS09.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS09.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS09.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.



LQS09.So.5 Invarianz

Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.10 Leistungseinheit LQS10 „Prüfungen der Spezifikationen von Komponenten und Modulen unterstützen“

3.7.10.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.1 Prüfbarkeit feststellen
(Prüfgegenstand
„SW-Entwurf.*SW-Komponenten-/SW-Modul-Beschreibung*“)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstand
„SW-Entwurf.*SW-Komponenten-/SW-Modul-Beschreibung*“)

Methode

– Statische Analyse (STAT)

3.7.10.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen die Prüfung von Komponenten und Modulen unterstützt wird.

3.7.10.3 Anforderungen

3.7.10.3.1 Anforderungen an die Schnittstellen

LQS10.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.10.3.2 Anforderungen an die Methodenunterstützung

LQS10.M.1 Statische Analyse (STAT)

LQS10.M.1.1 Pseudocode (PCODE)

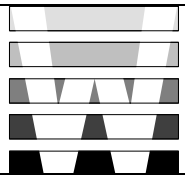
LQS10.M.1.1.1 Konsistenzprüfungen

LQS10.M.1.1.1.1 Konsistenzprüfungen von Datenelementen aus Datendeklarationen

LQS10.M.1.1.1.1.1 Unbenutzte Datenelemente
Es ist möglich, Datenelemente auflisten zu lassen, die von keiner Anweisung benutzt werden.

LQS10.M.1.1.1.1.2 Nicht initialisierte Datenelemente
Es ist möglich, Datenelemente auflisten zu lassen, die nicht initialisiert wurden.

LQS10.M.1.1.1.1.3 Nicht vereinbarte Datenelemente
Es ist möglich, Datenelemente auflisten zu lassen, die nicht explizit vereinbart wurden.



LQS10.M.1.1.1.1.4 Nicht ganzzahlige Indexdatenelemente
Es ist möglich, Indexdatenelemente auflisten zu lassen, die nicht ganzzahlig definiert sind.

LQS10.M.1.1.1.1.5 Ähnliche Datenelementnamen
Es ist möglich, Datenelemente auflisten zu lassen, die ähnliche Namen besitzen.

LQS10.M.1.1.1.2 Konsistenzprüfungen von Anweisungen für Berechnungen

LQS10.M.1.1.1.2.1 Nicht-arithmetische Datenelemente
Es ist möglich, Anweisungen für Berechnungen auflisten zu lassen, bei denen nicht-arithmetische Datenelemente verwendet werden.

LQS10.M.1.1.1.2.2 Verschiedene Datentypen
Es ist möglich, Anweisungen für Berechnungen auflisten zu lassen, bei denen Datenelemente mit verschiedenen Datentypen verwendet werden.

LQS10.M.1.1.1.2.3 Verschiedene Datenelementenlängen
Es ist möglich, Anweisungen für Berechnungen auflisten zu lassen, bei denen Datenelemente mit zwar gleichem Datentyp, aber mit verschiedener Längendefinition verwendet werden.

LQS10.M.1.1.1.3 Konsistenzprüfungen von Anweisungen für Zuweisungen

LQS10.M.1.1.1.3.1 Verschiedene Datentypen
Es ist möglich, Anweisungen für Zuweisungen auflisten zu lassen, bei denen Datenelemente mit verschiedenen Datentypen verwendet werden.

LQS10.M.1.1.1.3.2 Verschiedene Datenelementenlängen
Es ist möglich, Anweisungen für Zuweisungen auflisten zu lassen, bei denen Datenelemente mit zwar gleichem Datentyp, aber mit verschiedener Längendefinition verwendet werden.

LQS10.M.1.1.1.4 Konsistenzprüfungen von Anweisungen für Vergleiche

LQS10.M.1.1.1.4.1 Verschiedene Datentypen
Es ist möglich, Anweisungen für Vergleiche auflisten zu lassen, bei denen Datenelemente mit verschiedenen Datentypen verwendet werden.

LQS10.M.1.1.1.4.2 Verschiedene Datenelementenlängen
Es ist möglich, Anweisungen für Vergleiche auflisten zu lassen, bei denen Datenelemente mit zwar gleichem Datentyp, aber mit verschiedener Längendefinition verwendet werden.

LQS10.M.1.1.1.5 Konsistenzprüfungen von Kontrollstrukturen

LQS10.M.1.1.1.5.1 Syntaxcheck
Es ist möglich, die Syntax von Anweisungen automatisch prüfen zu lassen.

LQS10.M.1.1.1.5.2 Unerwünschte Kontrollstrukturen
Es ist möglich, die Verwendung unerwünschter Kontrollstrukturen auflisten zu lassen.

Unerwünschte Kontrollstrukturen können z. B. unbedingte Sprunganweisung („Goto“) oder dynamische Programmschalteranweisungen (wie „Alter“ in COBOL) sein.

LQS10.M.1.1.1.6 Konsistenzprüfungen von Schnittstellen

LQS10.M.1.1.1.6.1 Parameteranzahl
Es ist möglich, jeweils die aufrufenden und aufgerufenen SW-Komponenten

bzw. SW-Module auflisten zu lassen, bei denen die Anzahl der Parameter nicht übereinstimmt.

LQS10.M.1.1.1.6.2 Parametereigenschaften

LQS10.M.1.1.1.6.2.1 Verschiedene Datentypen
Es ist möglich, jeweils die aufrufenden und aufgerufenen SW-Komponenten bzw. SW-Module auflisten zu lassen, bei denen die Parameter mit verschiedenen Datentypen definiert wurden.

LQS10.M.1.1.1.6.2.2 Verschiedene Längen
Es ist möglich, jeweils die aufrufenden und aufgerufenen SW-Komponenten bzw. SW-Module auflisten zu lassen, bei denen die Parameter mit verschiedenen Längen definiert wurden.

LQS10.M.1.1.1.6.2.3 Parameter-Gebrauch
Es ist möglich, jeweils die aufrufenden und aufgerufenen SW-Komponenten bzw. SW-Module auflisten zu lassen, bei denen mit den Parametern nicht gearbeitet wird.

LQS10.M.1.1.1.7 Konsistenzprüfungen von Datenzugriffen

LQS10.M.1.1.1.7.1 Schließen von Dateien
Es ist möglich, die Dateien auflisten zu lassen, die vor Beendigung des SW-Komponenten- bzw. SW-Modulablaufs nicht geschlossen werden.

LQS10.M.1.1.1.7.2 Parameter bei CALL-Aufrufen
Es ist möglich, die CALL-Aufrufe aus SW-Modulen aufzulisten, bei denen die Art bzw. Zahl der CALL-Parameter nicht mit vorgegebenen Regeln übereinstimmt.

LQS10.M.1.1.2 Statistiken

LQS10.M.1.1.2.1 Kontrollstrukturen
Es ist möglich, ermitteln zu lassen, wie oft bestimmte Kontrollstrukturen (inklusive Abbruch- und Aufrufanweisungen) in einer SW-Komponente bzw. einem SW-Modul verwendet wurde.

LQS10.M.1.1.2.2 Maximale Schachtelungstiefe
Es ist möglich, die maximale Schachtelungstiefe von Kontrollstrukturen in einer SW-Komponente bzw. einem SW-Modul ermitteln zu lassen.

LQS10.M.1.1.2.3 Datei-/Datenbankzugriffe
Es ist möglich, die Art, Zahl und Ablaufzeit der Datei-/und Datenbankzugriffe je SW-Modul aufzulisten.

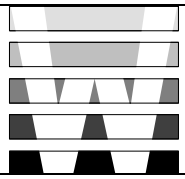
LQS10.M.1.1.3 Strukturanalysen

LQS10.M.1.1.3.1 Aufrufhierarchie
Es ist möglich, die Hierarchie der Aufrufe zwischen SW-Komponenten bzw. SW-Modulen darstellen zu lassen.

LQS10.M.1.1.3.2 SW-Komponenten-/-Modulstruktur
Es ist möglich, die interne Struktur von SW-Komponenten bzw. SW-Modulen darstellen zu lassen.

Unter interner Struktur von SW-Komponenten und SW-Modulen wird hier die Schachtelung in sich abgeschlossener Bestandteile einer Spezifikation (interne Prozeduren, Funktionen, Pakete) verstanden.

LQS10.M.1.1.4 Komplexitätsmessungen
Es ist möglich, die Komplexität innerhalb einer SW-Komponentee bzw. eines SW-Moduls ermitteln zu lassen.



Gängige Ansätze für die Messung von Softwarekomplexität sind die von McCabe bzw. Halstead.

LQS10.M.1.1.5 Konformität mit Standards
Es ist möglich, die Einhaltung von (Projekt-) Standards (SAA, CUA, Style Guides) zum Beispiel beim Maskenaufbau zu überprüfen.

LQS10.M.1.1.6 Prüfung von Konventionen

LQS10.M.1.1.6.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS10.M.1.1.6.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS10.M.1.1.6.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS10.M.1.1.6.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS10.M.1.1.6.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

LQS10.M.1.1.7 Anpassung
Individuell angepaßte Darstellungsformen des zu prüfenden Objektes werden innerhalb der Prüfung mit berücksichtigt.

LQS10.M.1.1.8 Positionierung bei Konsistenzproblemen
Aus den Auflistungen zu entdeckten Konsistenzproblemen ist eine Positionierung an die fragliche Stelle der Spezifikation möglich.

LQS10.M.1.1.9 Vollständigkeit der Beschreibung
Es ist möglich, sich die Objekte auflisten zu lassen, zu denen keine Beschreibungen existieren.

3.7.10.3.3 Anforderungen an die Funktionen

LQS10.F.1 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

Unter wählbaren Merkmalen sind hier u. a. Konsistenzprüfungen von Datenelementen, Berechnungen, Zuweisungen und Vergleichsanweisungen, Komplexitätsmessungen, Verwendung von Sprachmitteln und Kommentierung aufzufassen.

LQS10.F.2 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS10.F.3 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.10.3.4 Sonstige Anforderungen

LQS10.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS10.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS10.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS10.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS10.So.5 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.11 Leistungseinheit LQS11 „Prüfungen der logischen Schemaspezifikationen unterstützen“

3.7.11.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- QS 4.1 Prüfbarkeit feststellen
  (Prüfgegenstände
  „SW-Entwurf.*Datenbankbeschreibung*“ und
  „Datenkatalog“)
- QS 4.2 Produkt inhaltlich prüfen
  (Prüfgegenstände
  „SW-Entwurf.*Datenbankbeschreibung*“ und
  „Datenkatalog“)

Methode

- Statische Analyse (STAT)

3.7.11.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen überprüft werden kann,

- ob die logische Schemaspezifikation alle Elemente der konzeptionellen Schemaspezifikation enthält,
- ob die Integritätsbedingungen der konzeptionellen Schemaspezifikation in die entsprechende logische Schemaspezifikation übernommen wurden.

3.7.11.3 Anforderungen

3.7.11.3.1 Anforderungen an die Schnittstellen

LQS11.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.11.3.2 Anforderungen an die Methodenunterstützung

LQS11.M.1 Statische Analyse (STAT)

LQS11.M.1.1 Data Navigation Modelling (DNAV)/Logische DB-Modellierung (LOGM)

LQS11.M.1.1.1 Prüfung von Konventionen

LQS11.M.1.1.1.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS11.M.1.1.1.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS11.M.1.1.1.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS11.M.1.1.1.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS11.M.1.1.1.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

LQS11.M.1.1.2 Fehlende Elemente

LQS11.M.1.1.2.1 In logischen Schemaspezifikationen fehlende Elemente
Es ist möglich, alle Entitätstypen, Beziehungstypen und Attribute der konzeptionellen Schemaspezifikationen auflisten zu lassen, die nicht in logische Schemaspezifikationen übernommen wurden.

LQS11.M.1.1.2.2 Nicht vereinbarte Bestandteile in logischen Schemaspezifikationen
Es ist möglich, anwenderspezifische Bestandteile in logischen Schemaspezifikationen, die nicht in den konzeptionellen Schemaspezifikationen enthalten sind, auflisten zu lassen.

LQS11.M.1.1.3 Übernahme von Integritätsanforderungen bei Transformationen
Es ist möglich, die bei der Transformation von der konzeptionellen Schemaspezifikation in die logische Schemaspezifikation nicht übernommenen Integritätsbedingungen auflisten zu lassen.

LQS11.M.1.1.4 Datennavigations-Diagramme
Es ist möglich, die Datennavigations-Diagramme auflisten zu lassen, die bei der Transformation der konzeptionellen Schemaspezifikation in die logische Schemaspezifikation nicht berücksichtigt wurden.

LQS11.M.1.1.5 Vollständigkeit der Beschreibung
Es ist möglich, sich die Objekte auflisten zu lassen, zu denen keine Beschreibungen existieren.

LQS11.M.1.1.6 Anpassung
Individuell angepaßte Darstellungsformen des zu prüfenden Objektes werden innerhalb der Prüfung mit berücksichtigt.

3.7.11.3.3 Anforderungen an die Funktionen

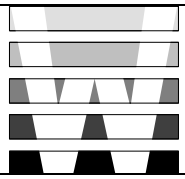
LQS11.F.1 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

LQS11.F.2 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS11.F.3 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.



3.7.11.3.4 Sonstige Anforderungen

LQS11.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS11.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS11.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS11.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS11.So.5 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.12 Leistungseinheit LQS12 „Code statisch prüfen“

3.7.12.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- SE 6.3-SW Selbstprüfung des SW-Moduls/der Datenbank durchführen (Prüfgegenstand „SW-Modul“)
- SE 7.2-SW Selbstprüfung der SW-Komponente durchführen (Prüfgegenstand „SW-Komponente“)
- SE 7.4-SW Selbstprüfung der SW-Einheit durchführen (Prüfgegenstand „SW-Einheit“)
- QS 4.1 Prüfbarkeit feststellen (Prüfgegenstände „SW-Modul“, „SW-Komponente“ und „SW-Einheit“)
- QS 4.2 Produkt inhaltlich prüfen (Prüfgegenstände „SW-Modul“, „SW-Komponente“ und „SW-Einheit“)

Methode

- Statische Analyse (STAT)

3.7.12.2 Kurzcharakteristik

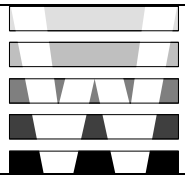
In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen Programm-Quellcode in bezug auf Form und Struktur analysiert wird, ohne daß das Programm ausgeführt werden muß. Es sollen hierbei

- die Einhaltung bestimmter Programmier-Richtlinien untersucht werden,
- syntaktisch/semantische Prüfungen stattfinden, die über die Analyse des Compilers hinausgehen können,
- eine Aussage über die Qualität der Software getroffen werden (Software-Metriken).

3.7.12.3 Anforderungen

3.7.12.3.1 Anforderungen an die Schnittstellen

LQS12.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.



3.7.12.3.2 Anforderungen an die Methodenunterstützung

LQS12.M.1 Statische Analyse (STAT)

LQS12.M.1.1 Prüfung von Konventionen

LQS12.M.1.1.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS12.M.1.1.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS12.M.1.1.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS12.M.1.1.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS12.M.1.1.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

LQS12.M.1.2 Lexikalische Analyse

LQS12.M.1.2.1 Funktionsübersicht
Es ist möglich, die in einem Prüfgegenstand definierten Funktionen in einer Übersicht anzuzeigen.

Die Übersicht enthält mindestens die Kurzbeschreibung der Definition (Name, Rückgabe-Typ, Parameter und deren Typen) sowie die Beschreibung der Fundstelle mit zugehörigem Dateinamen.

LQS12.M.1.2.2 Querverweise
Es ist möglich, eine Querverweisliste (Cross Reference) anzulegen, die Auskunft über alle Verwendungen und Deklarationen von Variablen und Funktionen gibt.

LQS12.M.1.3 Allgemeine syntaktische Analyse

LQS12.M.1.3.1 Undefiniert
Es ist möglich, verwendete aber nicht deklarierte Variable auflisten zu lassen.

LQS12.M.1.3.2 Uninitialisiert
Es ist möglich, verwendete und deklarierte aber nicht initialisierte Variable auflisten zu lassen.

LQS12.M.1.3.3 Unbenutzt
Es ist möglich, deklarierte aber nicht verwendete Variable auflisten zu lassen.

LQS12.M.1.3.4 Typfehler
Es ist möglich, Variable, die anders gebraucht wurden, als sie vom Typ her deklariert wurden, auflisten zu lassen.

Dies muß eine Warnung, nicht notwendig einen Fehler erzeugen.

LQS12.M.1.3.5 Typumwandlung
Es ist möglich, sprach- oder auch betriebssystemabhängige Typen auflisten zu lassen.

Z. B. die Behandlung von Zeigern als Integer, oder die Zuweisung Long nach Short.

LQS12.M.1.3.6 Toter Code
Es ist möglich, bei einer Auflistung des Prüfgegenstandes nicht erreichbare Anweisungen auflisten zu lassen.

LQS12.M.1.3.7 Parameteranzahl
Es ist möglich, Funktionsaufrufe auflisten zu lassen, bei denen die Anzahl der übergebenen Parameter nicht mit der Deklaration übereinstimmt.

Dies muß eine Warnung, nicht notwendig einen Fehler erzeugen.

LQS12.M.1.3.8 Parametertypen
Es ist möglich, Funktionsaufrufe auflisten zu lassen, bei denen die Parametertypen nicht mit der Deklaration übereinstimmen.

Dies muß eine Warnung, nicht notwendig einen Fehler erzeugen.

LQS12.M.1.3.9 Funktionstyp
Es ist möglich, die Funktionen auflisten zu lassen, bei denen kein resultierender Funktionstyp deklariert wurde.

LQS12.M.1.3.10 Rückkehr
Es ist möglich, deklarierte nicht typfreie Funktionstypen aufzulisten, von denen kein Wert oder ein Wert falschen Typs zurückgegeben wurde.

Dies muß eine Warnung, nicht notwendig einen Fehler erzeugen. Ein typfreier Funktionstyp wäre in C beispielsweise der Typ „void“.

LQS12.M.1.3.11 Ignorieren
Es ist möglich, Funktionen aufzulisten, bei denen deklarierte Rückgabewerte vom aufrufenden Sprachkonstrukt nicht ausgewertet wurden.

Dies muß eine Warnung, nicht notwendig einen Fehler erzeugen.

LQS12.M.1.3.12 Unbenutzt
Es ist möglich, Funktionen aufzulisten, die deklarierte und übergebene Parameter nicht benutzen.

Dies muß eine Warnung, nicht notwendig einen Fehler erzeugen.

LQS12.M.1.3.13 Ablaufstruktur

LQS12.M.1.3.13.1 Endlosschleifen
Es ist möglich, Endlosschleifen zu erkennen.

LQS12.M.1.3.13.2 Sprünge
Es ist möglich, unerwünschte Sprünge zu erkennen.

Derartige Sprünge sind z. B. Einsprünge in Schleifen.

LQS12.M.1.4 Strukturelle Analyse

LQS12.M.1.4.1 Funktionsgraph
Es ist möglich, die Aufrufhierarchie der Funktionen als Graph darzustellen.

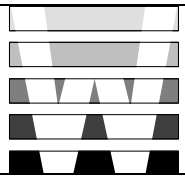
LQS12.M.1.4.2 Aufrufhierarchie
Für jede Funktion ist erkennbar, welche anderen Funktionen sie aufruft.

LQS12.M.1.4.3 Benutzung
Für jede Funktion ist erkennbar, von welchen anderen Funktionen sie aufgerufen wird.

LQS12.M.1.5 Erweiterte lexikalische und strukturelle Analyse

LQS12.M.1.5.1 Importübersicht
Es ist möglich, eine Übersicht über alle in ein Modul importierten Funktionen und Variablen zu erzeugen.

LQS12.M.1.5.2 Exportübersicht
Es ist möglich, eine Übersicht über alle aus einem Modul exportierten Funktionen und Variablen zu erzeugen.



LQS12.M.1.5.3 Referenzübersicht
Es ist möglich, zu jedem Modul eine Übersicht aller externen Referenzen zu erzeugen, aus der die Aufrufstelle bzw. Aufruffolge erkennbar ist.

LQS12.M.1.5.4 Modultiefe
Es ist möglich, für jedes Modul die maximale Schachtelungstiefe zu berechnen.

LQS12.M.1.5.5 Funktionstiefe
Es ist möglich, für jede Funktion die maximale Schachtelungstiefe zu berechnen.

LQS12.M.1.6 Software-Metriken

LQS12.M.1.6.1 Codezeilen
Es ist möglich, die Anzahl der Codezeilen (LOC: Lines of Code) zu ermitteln.

LQS12.M.1.6.2 Anweisungen
Es ist möglich, die Anzahl der Anweisungen zu ermitteln.

LQS12.M.1.6.3 Codelänge
Es ist möglich, die durchschnittliche Länge einer Anweisung bzw. Funktion in LOC zu ermitteln.

LQS12.M.1.6.4 Sprunganzahl
Es ist möglich, die Anzahl unbedingter Sprünge und Abbrüche zu ermitteln.

Z. B. Gotos oder Breaks, Exits.

LQS12.M.1.6.5 Kommentarhäufigkeit
Es ist möglich, die Häufigkeit von Programm-Kommentaren zu ermitteln.

LQS12.M.1.6.6 Kommentaranteil
Es ist möglich, den Anteil Kommentarzeilen am Gesamtumfang der LOC zu ermitteln.

LQS12.M.1.6.7 Schlüsselworthäufigkeit
Es ist möglich, die Häufigkeiten der verschiedenen Schlüsselwörter der Sprache zu ermitteln.

LQS12.M.1.6.8 Aufrufzahl
Es ist möglich, die lineare Aufrufanzahl von Funktionen zu berechnen.

LQS12.M.1.6.9 Referenzzahl
Es ist möglich, die Anzahl der Referenzen für jede Variable zu ermitteln.

LQS12.M.1.6.10 Komplexität
Es ist möglich, eine Maßzahl zur Komplexität einzelner Module/Funktionen zu errechnen.

Diese Maßzahl sollte z. B. typische Abweichungen zur durchschnittlichen Komplexität identifizieren (Einzeiler-Funktionen, mehrfach rekursive Funktionen, auffällig lange Funktionen).

LQS12.M.1.6.10.1 Es ist möglich, die Metriken als Tabellen oder Grafik darzustellen.

LQS12.M.1.6.10.2 Es ist möglich, Sollgrößen zu definieren und Abweichungen kenntlich zu machen.

LQS12.M.1.7 Anpassung
Individuell angepaßte Darstellungsformen des zu prüfenden Objektes werden innerhalb der Prüfung mit berücksichtigt.

3.7.12.3.3 Anforderungen an die Funktionen

LQS12.F.1 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

LQS12.F.2 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS12.F.3 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.12.3.4 Sonstige Anforderungen

LQS12.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS12.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS12.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS12.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS12.So.5 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.13 Leistungseinheit LQS13 „Texte statisch prüfen (Text ungleich Code)“

3.7.13.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- QS 4.1 Prüfbarkeit feststellen
  (alle Prüfgegenstände)
- QS 4.2 Produkt inhaltlich prüfen
  (alle Prüfgegenstände)

Methode

- Statische Analyse (STAT)

3.7.13.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen

- Texte hinsichtlich korrekter Syntax und Schreibweisen analysiert und
- Syntaxfehler und falsche Schreibweisen automatisch korrigiert werden können.

werden können.

3.7.13.3 Anforderungen

3.7.13.3.1 Anforderungen an die Schnittstellen

LQS13.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.13.3.2 Anforderungen an die Methodenunterstützung

LQS13.M.1 Statische Analyse (STAT)

LQS13.M.1.1 Prüfung von Konventionen

LQS13.M.1.1.1 Projektstandards
Es ist möglich, die abzuprüfenden Konventionen projektspezifisch festzulegen.

LQS13.M.1.1.2 Namenskonventionen
Es ist möglich, Namenskonventionen abzuprüfen.

LQS13.M.1.1.3 Beschränkungen
Es ist möglich, Größen- und Komplexitätsbeschränkungen abzuprüfen.

LQS13.M.1.1.4 Layoutkonventionen
Es ist möglich, Layoutkonventionen abzuprüfen.

LQS13.M.1.1.5 Darstellungsformen
Es ist möglich, Festlegungen auf bestimmte Darstellungsformen abzuprüfen.

LQS13.M.1.2 Rechtschreibprüfung

LQS13.M.1.2.1 Deutsches Wörterbuch
Es existiert ein deutsches Wörterbuch für die Rechtschreibprüfung.

LQS13.M.1.2.2 Englisches Wörterbuch
Es existiert ein englisches Wörterbuch für die Rechtschreibprüfung.

LQS13.M.1.2.3 Nutzerspezifisches Wörterbuch
Neben dem Stammwörterbuch wird ein nutzerspezifisches Wörterbuch für die Rechtschreibprüfung zur Verfügung gestellt.

LQS13.M.1.2.4 Erweiterbarkeit
Das nutzerspezifische Wörterbuch für die Rechtschreibprüfung ist erweiterbar.

LQS13.M.1.2.5 Getrennte Wörter
Getrennte Wörter werden als solche erkannt.

LQS13.M.1.2.6 Sprachspezifische Trennung
Die Worttrennung erfolgt gemäß den Trennungsregeln der jeweiligen Sprache.

Beispiele: Das Wort „Schiffahrt“ wird bei der Trennung zu „Schiff-fahrt“, „Drucker“ wird zu „Druk-ker“ (bei Anwendung der Regeln, wie sie vor *der Rechtschreibreform der deutschen Sprache gegolten haben).*

LQS13.M.1.3 Anpassung
Individuell angepaßte Darstellungsformen des zu prüfenden Objektes werden innerhalb der Prüfung mit berücksichtigt.

3.7.13.3.3 Anforderungen an die Funktionen

LQS13.F.1 Begrenzung der Prüfung
Es ist möglich, Umfang und Ausmaß der Prüfung auf wählbare Merkmale einzugrenzen.

LQS13.F.2 Prozedursprache
Über den Einsatz einer Prozedursprache ist es möglich, weitere individuelle Prüfkriterien zu berücksichtigen.

Eine Prozedursprache ermöglicht die automatisierte Abprüfung weiterer individueller Prüfkriterien.

LQS13.F.3 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

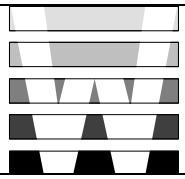
Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.13.3.4 Sonstige Anforderungen

LQS13.So.1 Grafik
Grafische Anteile im zu prüfenden Text werden ignoriert.

LQS13.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS13.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.



LQS13.So.4 Hintergrund

Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS13.So.5 Endesignal

Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS13.So.6 Invarianz

Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.14 Leistungseinheit LQS14 „Prüffälle generieren“

3.7.14.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 2.3 Prüffälle festlegen („Prüfspezifikation.*Prüffälle*“)

Methode

– Black-box-Testfallentwurf (BBTE)

– White-box-Testfallentwurf (WBTE)

3.7.14.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen Prüffälle generiert und verwaltet werden können.

3.7.14.3 Anforderungen

3.7.14.3.1 Anforderungen an die Schnittstellen

LQS14.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LQS14.S.2 Eingangs-Schnittstelle zu LSE02 „Spezifikation von Nutzerschnittstellen unterstützen“
Es ist möglich, mit LSE02 erstellte Spezifikationen für Schnittstellen ohne weitere Transformation aus der Objektverwaltung zur Generierung von Prüffällen auszuwerten.

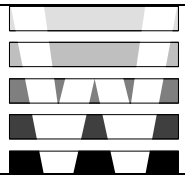
LQS14.S.3 Eingangs-Schnittstelle zu LSE04 „Prozeßmodellierung unterstützen“
Es ist möglich, mit LSE04 erstellte Prozeßmodelle ohne weitere Transformation aus der Objektverwaltung zur Generierung von Prüffällen auszuwerten.

LQS14.S.4 Eingangs-Schnittstelle zu LSE06 „Spezifikation der Funktionen unterstützen“
Es ist möglich, mit LSE06 erstellte Spezifikationen für Funktionen ohne weitere Transformation aus der Objektverwaltung zur Generierung von Prüffällen auszuwerten.

LQS14.S.5 Eingangs-Schnittstelle zu LSE07 „Modellierung der Informationsflüsse unterstützen“
Es ist möglich, mit LSE07 erstellte Informationsflußdiagramme ohne weitere Transformation aus der Objektverwaltung zur Generierung von Prüffällen auszuwerten.

LQS14.S.6 Eingangs-Schnittstelle zu LSE08 „Modellierung der Funktionsdynamik unterstützen“
Es ist möglich, mit LSE08 erstellte Ablaufmodelle ohne weitere Transformation aus der Objektverwaltung zur Generierung von Prüffällen auszuwerten.

LQS14.S.7 Eingangs-Schnittstelle zu LSE10 „Spezifikation der Komponenten und Module unterstützen“
Es ist möglich, mit LSE10 erstellte Spezifikationen für Komponenten und Module ohne



weitere Transformation aus der Objektverwaltung zur Generierung von Prüffällen auszuwerten.

LQS14.S.8 Eingangs-Schnittstelle zu LSE11 „Spezifikation der Datenbanken unterstützen“
Es ist möglich, mit LSE11 erstellte logische Schemaspezifikationen für Datenbanken ohne weitere Transformation aus der Objektverwaltung zur Generierung von Prüffällen auszuwerten.

LQS14.S.9 Eingangs-Schnittstelle zu LSE12 „Komponenten und Module generieren“
Es ist möglich, mit LSE12 generierten Quellcode für SW-Komponenten und SW-Module ohne weitere Transformation aus der Objektverwaltung zur Generierung von Prüffällen auszuwerten.

3.7.14.3.2 Anforderungen an die Methodenunterstützung

LQS14.M.1 Black-box-Testfallentwurf (BBTE)

LQS14.M.1.1 Zulässige Eingabedaten

LQS14.M.1.1.1 Normalwerte aus Kontrollinformationen
Es ist möglich, aus Kontrollinformationen Normalwerte von zulässigen Eingabedaten zu generieren.

Kontrollinformationen der Prozeßmodelle sind u. a. Ereignisse, Bedingungen, Zustände, Signale und zeitliche Reihenfolgen von Ereignissen, Bedingungen, Zuständen und Signalen. Kontrollinformationen von Datenfluß-Diagrammen sind u. a. die Auslöser von Datenflüssen. Kontrollinformationen der Ablaufmodelle sind Ereignisse, Bedingungen und zeitliche Reihenfolgen von Ereignissen und Bedingungen.

LQS14.M.1.1.2 Grenzwerte aus Kontrollinformationen
Es ist möglich, aus Kontrollinformationen Grenzwerte von zulässigen Eingabedaten zu generieren.

Unter Grenzwerten sind hier Rand- und Extremwerte aufzufassen.

LQS14.M.1.1.3 Normalwerte aus Datendeklarationen
Es ist möglich, aus Datendeklarationen Normalwerte von zulässigen Eingabedaten zu generieren.

Hierbei werden Datentypen und Wertebereiche der Datenelemente aus den Datendeklarationen ausgewertet.

LQS14.M.1.1.4 Grenzwerte aus Datendeklarationen
Es ist möglich, aus Datendeklarationen Grenzwerte von zulässigen Eingabedaten zu generieren.

Unter Grenzwerten sind hier Rand- und Extremwerte aufzufassen.

LQS14.M.1.2 Formal unzulässige Eingabedaten

LQS14.M.1.2.1 Kontrollinformationen
Es ist möglich, aus Kontrollinformationen formal unzulässige Eingabedaten zu generieren.

Formal unzulässige Daten müssen vom Prüfgegenstand durch Plausibilitätsprüfungen erkannt und bearbeitet werden.

LQS14.M.1.2.2 Datendeklarationen
Es ist möglich, aus Datendeklarationen formal unzulässige Eingabedaten zu generieren.

Formal unzulässige Daten müssen vom Prüfgegenstand durch Plausibilitätsprüfungen erkannt und bearbeitet werden.

LQS14.M.1.3 Fachlich unzulässige Eingabedaten

LQS14.M.1.3.1 Normalwerte aus Kontrollinformationen
Es ist möglich, aus Kontrollinformationen Normalwerte von fachlich unzulässigen Eingabedaten zu generieren.

Fachlich unzulässige Normalwerte müssen vom Prüfgegenstand durch Plausibilitätsprüfungen erkannt und bearbeitet werden.

LQS14.M.1.3.2 Grenzwerte aus Kontrollinformationen
Es ist möglich, aus Kontrollinformationen Grenzwerte von fachlich unzulässigen Eingabedaten zu generieren.

Unter „Grenzwerte“ sind hier Rand- und Extremwerte aufzufassen. Fachlich unzulässige Grenzwerte müssen vom Prüfgegenstand durch Plausibilitätsprüfungen erkannt und bearbeitet werden.

LQS14.M.1.3.3 Normalwerte aus Datendeklarationen
Es ist möglich, aus Datendeklarationen Normalwerte von fachlich unzulässigen Eingabedaten zu generieren.

Fachlich unzulässige Normalwerte müssen vom Prüfgegenstand durch Plausibilitätsprüfungen erkannt und bearbeitet werden.

LQS14.M.1.3.4 Grenzwerte aus Datendeklarationen
Es ist möglich, aus Datendeklarationen Grenzwerte von fachlich unzulässigen Eingabedaten zu generieren.

Unter Grenzwerten sind hier Rand- und Extremwerte aufzufassen. Fachlich unzulässige Grenzwerte müssen vom Prüfgegenstand durch Plausibilitätsprüfungen erkannt und bearbeitet werden.

LQS14.M.1.4 Intuitive Prüffallermittlung
Es ist möglich, die generierten Prüffälle um vom Prüfer intuitiv ermittelte Prüffälle manuell zu ergänzen.

LQS14.M.1.5 Prüffallkombinationen

LQS14.M.1.5.1 Zulässige Eingabedaten
Es ist möglich, Kombinationen zulässiger Eingabedaten zu generieren.

LQS14.M.1.5.2 Unzulässige Eingabedaten
Es ist möglich, Kombinationen unzulässiger Eingabedaten zu generieren.

LQS14.M.1.5.3 Zulässige und unzulässige Eingabedaten
Es ist möglich, Kombinationen zulässiger und unzulässiger Eingabedaten zu generieren.

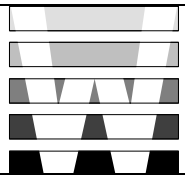
LQS14.M.1.6 Erwartete Ausgabedaten
Es ist möglich, für die Eingabedaten die erwarteten Ausgabedaten bzw. Reaktionsweisen manuell zu erfassen.

LQS14.M.1.7 Prüfdatenbestand
Es ist möglich, aus den logischen Schemaspezifikationen für Datenbanken und Datenstrukturen Daten für Prüfdateien bzw. Prüfdatenbanken zu generieren.

Hierbei werden Datentypen und Wertebereiche der Datenelemente aus den logischen Schemaspezifikationen für Datenbanken und Datenstrukturen ausgewertet.

LQS14.M.2 White-box-Testfallentwurf (WBTE)

LQS14.M.2.1 Pfadabdeckung
Es ist möglich, Eingabedaten zu definieren, mit denen eine geforderte Mindestanzahl von Anweisungen im Prüfgegenstand mindestens einmal zur Ausführung gebracht werden kann.



LQS14.M.2.2 Anweisungsabdeckung
Es ist möglich, Eingabedaten zu definieren, mit denen eine geforderte Mindestanzahl von Anweisungen im Prüfgegenstand mindestens einmal zur Ausführung gebracht werden kann.

LQS14.M.2.3 Zweigabdeckung
Es ist möglich, Eingabedaten zu definieren, mit denen eine geforderte Mindestanzahl von Zweigen im Prüfgegenstand mindestens einmal zur Ausführung gebracht werden kann.

LQS14.M.2.4 Bedingungsabdeckung
Es ist möglich, Eingabedaten zu definieren, mit denen eine geforderte Mindestanzahl von Bedingungen im Prüfgegenstand mindestens einmal zur Ausführung gebracht werden kann.

LQS14.M.2.5 Zweig-/Bedingungsabdeckung
Es ist möglich, Eingabedaten zu definieren, mit denen eine geforderte Mindestanzahl von Zweigen und Bedingungen im Prüfgegenstand mindestens einmal zur Ausführung gebracht werden kann.

LQS14.M.2.6 Abdeckung aller Mehrfachbedingungen
Es ist möglich, Eingabedaten zu definieren, mit denen eine geforderte Mindestanzahl aller möglichen Kombinationen der Bedingungen einer Entscheidung im Prüfgegenstand mindestens einmal zur Ausführung gebracht werden kann.

LQS14.M.2.7 Erwartete Ausgabedaten
Es ist möglich, für die Eingabedaten die erwarteten Ausgabedaten manuell zu erfassen.

3.7.14.3.3 Anforderungen an die Funktionen

LQS14.F.1 Prüffallverwaltung

LQS14.F.1.1 Kommandodatei
Es ist möglich, Prüffälle in einer Kommandodatei zu speichern.
Ein Prüffall umfaßt die Eingabedaten und die erwarteten Ausgabedaten.

LQS14.F.1.2 Ergänzung der Kommandodatei

LQS14.F.1.2.1 Ergänzung von Prüffällen
Es ist möglich, eine Kommandodatei ab einer definierbaren Stelle um weitere Prüffälle zu ergänzen.

LQS14.F.1.2.2 Ergänzung von Kommentaren
Es ist möglich, eine Kommandodatei um Kommentare zur Erläuterung von Prüffällen zu ergänzen.

LQS14.F.1.2.3 Auswerten der Kommentare
Es ist möglich, die Kommentare in einer Kommandodatei so zu führen, daß sie als Erläuterungen zu Prüffällen während des Prüflaufs eingeblendet werden können.

LQS14.F.1.3 Kombinieren von Kommandodateien
Es ist möglich, Kommandodateien für Prüfläufe zu kombinieren.

LQS14.F.2 Prüffallaufzeichung

LQS14.F.2.1 Tastenanschläge
Es ist möglich, Tastenanschläge protokollieren zu lassen.

LQS14.F.2.2 Mausbewegungen
Es ist möglich, Mausbewegungen protokollieren zu lassen.

LQS14.F.2.3 Maustastenbetätigung
Es ist möglich, das Betätigen der unterschiedlichen Maustasten protokollieren zu lassen.

3.7.14.3.4 Sonstige Anforderungen

LQS14.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS14.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.7.15 Leistungseinheit LQS15 „Dynamische Prüfungen unterstützen“

3.7.15.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstände
„Spezifikationen von Schnittstellen (LSE02)“,
„Prozeßmodelle“ (LSE04),
„Spezifikationen von Funktionen (LSE06)“,
„Informationsflußmodelle (LSE07)“,
„Ablaufmodelle (LSE08)“,
„Spezifikationen von Komponenten und Modulen (LSE10)“,
„SW-Modul“, „SW-Komponente“, „Datenbank“, „SW-Einheit“, „Segment“ und „System“)

Methode

– Testen (T)

3.7.15.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen

– dynamische Prüfungen von Prüfgegenständen anhand festgelegter Prüffälle durchgeführt werden können,
– Abweichungen zwischen den tatsächlichen und den erwarteten Ergebnissen der Prüfläufe gefunden werden können,
– die Wirksamkeit der Prüffälle eingeschätzt werden kann.

3.7.15.3 Anforderungen

3.7.15.3.1 Anforderungen an die Schnittstellen

LQS15.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.15.3.2 Anforderungen an die Methodenunterstützung

LQS15.M.1 Testen (T)

LQS15.M.1.1 Navigation

LQS15.M.1.1.1 Startpunkt
Es ist möglich, den Startpunkt einer Navigation festzulegen.
Bei einem Informationsflußmodell ist beispielsweise ein bestimmter Informationsfluß der Startpunkt.

LQS15.M.1.1.2 Verzweigung
Es ist möglich, bei Verzweigungen zu entscheiden, welcher Navigationsweg weiter verfolgt werden soll.

LQS15.M.1.1.3 Terminierung
Es ist möglich, bei Terminierung eines Navigationsweges zu entscheiden, ob und welcher Navigationsweg weiter verfolgt werden soll.

LQS15.M.1.1.4 Navigationsrichtung
Es ist möglich, zwischen vorwärts oder rückwärts gerichteter Navigation auszuwählen.

LQS15.M.1.2 Prüflaufausführung

LQS15.M.1.2.1 Abbruch eines Prüflaufs
Es ist möglich, einen Prüflauf durch Abbruch abzubrechen.

LQS15.M.1.2.2 Unterbrechen eines Prüflaufs
Es ist möglich, einen Prüflauf zu unterbrechen.

LQS15.M.1.2.3 Wiederaufnahme eines Prüflaufs
Es ist möglich, einen unterbrochenen Prüflauf an der Unterbrechungsstelle wieder aufzunehmen.

LQS15.M.1.2.4 Externe Referenzen
Es ist möglich, externe Referenzen zu befriedigen.

Durch eine Befriedigung externer Referenzen ist es beispielsweise möglich, ein einzelnes Modul auszuführen.

LQS15.M.1.3 Ergebnisanalyse

LQS15.M.1.3.1 C0-Abdeckung
Es ist möglich, die C0-Abdeckung (Objektabdeckung) zu berechnen.

Im Falle von Code sind unter Objekten Anweisungen zu verstehen. Die C0-Abdeckung ist das Verhältnis der durchlaufenen Objekte zur Gesamtzahl der Objekte des Prüfgegenstandes.

LQS15.M.1.3.2 C0-Abdeckungsergebnis
Es ist möglich, die nicht durchlaufenen Objekte auflisten zu lassen.

Im Falle von Code sind unter Objekten Anweisungen zu verstehen.

LQS15.M.1.3.3 C1-Abdeckung
Es ist möglich, die C1-Abdeckung (Zweigabdeckung) zu berechnen.

Die C1-Abdeckung ist das Verhältnis der durchlaufenen Zweige zur Gesamtzahl der möglichen Zweige des Prüfgegenstandes.

LQS15.M.1.3.4 C1-Abdeckungsergebnis
Es ist möglich, die nicht durchlaufenen Zweige auflisten zu lassen.

LQS15.M.1.4 Prüffallzuordnung
Eine eindeutige Zuordnung der Prüffälle (einschließlich der erwarteten Prüfergebnisse) aus LQS14 und den dynamischen Prüfungen sowie den erzielten Prüfergebnissen muß sichergestellt sein.

LQS15.M.1.5 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

LQS15.M.1.6 Prüfauswertung

LQS15.M.1.6.1 Prüfungsauswertung
Eine automatisierte Prüfungsauswertung ist möglich.

Die Prüfungsauswertung erfolgt durch eine Gegenüberstellung der erwarteten zu den erzielten Ergebnissen.

LQS15.M.1.6.2 Fehlerhafte Prüffälle
Es ist möglich, fehlerhafte Prüffälle auflisten zu lassen.

Bei jedem fehlerhaften Prüffall werden neben den Ein-/Ausgabedaten auch die erwarteten Ausgabedaten aufgelistet.

3.7.15.3.3 Anforderungen an die Funktionen

LQS15.F.1 Prüffallverwaltung

LQS15.F.1.1 Kommandodatei
Es ist möglich, die Prüffälle aus LQS14 in Form von Kommandodateien verarbeiten zu lassen.

Ein Prüffall umfaßt die Eingabedaten und die erwarteten Ausgabedaten.

LQS15.F.2 Regressionsprüfungen

LQS15.F.2.1 Prüfungsautomatisierung
Es ist möglich, bei einer Prüfung die Interaktionen des Prüfers abzuspeichern, so daß bei einer Regressionsprüfung der Prüfer nur mehr überwachend, nicht aber ausführend tätig werden muß.

LQS15.F.2.2 Prüfungsvergleich
Ein Vergleich der Prüfungsergebnisse zweier Prüfungen ist automatisch möglich.

3.7.15.3.4 Sonstige Anforderungen

LQS15.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LQS15.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS15.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS15.So.4 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS15.So.5 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS15.So.6 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.16 Leistungseinheit LQS16 „Konsistenz des Ausschreibungsrelevanten Tailoring sichern“

3.7.16.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstand
„Projekthandbuch.*Projektspezifisches V-Modell*“)

Methode

keine

3.7.16.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen geprüft werden kann, ob mit durchgeführten Streichungen von Aktivitätentypen und Produkttypen aus dem allgemeinen V-Modell ein konsistentes projektspezifisches V-Modell entstanden ist.

3.7.16.3 Anforderungen

3.7.16.3.1 Anforderungen an die Schnittstellen

LQS16.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.16.3.2 Anforderungen an die Methodenunterstützung

keine

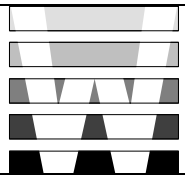
3.7.16.3.3 Anforderungen an die Funktionen

LQS16.F.1 Überflüssige Aktivitätentypen
Es ist möglich zu prüfen, ob alle Aktivitätentypen, die keine Produkttypen erzeugen oder weiter bearbeiten, gestrichen wurden.

Ein Produkttyp wird von einem Aktivitätentyp erzeugt, wenn er im Produktfluß dieser Aktivität eingangsseitig im Zustand „geplant“ und ausgangsseitig in einem Zustand verschieden von „geplant“ auftritt.

LQS16.F.2 Überflüssige Produkttypen
Es ist möglich zu prüfen, ob alle Produkttypen, die nirgends benötigt werden, gestrichen wurden.

Ein Produkttyp wird von einem Aktivitätentyp benötigt, wenn er im Produktfluß dieses Aktivitätentyps eingangsseitig auftritt.



LQS16.F.3 Abwicklungstexte

LQS16.F.3.1 Aktivitätentypen
Es ist möglich zu prüfen, ob der Abwicklungstext von nicht gestrichenen Aktivitätentypen keine Verweise auf bereits gestrichene Aktivitätentypen enthält.

LQS16.F.3.2 Produkttypen
Es ist möglich zu prüfen, ob der Abwicklungstext von nicht gestrichenen Aktivitätentypen keine Verweise auf bereits gestrichene Produkttypen enthält.

LQS16.F.4 Streichbedingungen
Es ist möglich zu prüfen, ob alle Ausschreibungsrelevanten Streichbedingungen, die zum projektspezifischen V-Modell führen, berücksichtigt wurden.

LQS16.F.5 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.16.3.4 Sonstige Anforderungen

LQS16.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS16.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS16.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS16.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS16.So.5 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.17 Leistungseinheit LQS17 „Prüfungen der Projektressourcenverwaltung unterstützen“

3.7.17.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstand
„Projektplan.*Einsatzmittelplan*“)

Methode

keine

3.7.17.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen geprüft werden kann,

– ob für alle Ressourcen die Tarife und Kalender initialisiert wurden,
– ob jedem Mitarbeiter eine Rolle aus dem V-Modell zugeordnet wurde und
– ob jede Rolle aus dem V-Modell von einem Mitarbeiter übernommen wird.

3.7.17.3 Anforderungen

3.7.17.3.1 Anforderungen an die Schnittstellen

LQS17.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.17.3.2 Anforderungen an die Methodenunterstützung

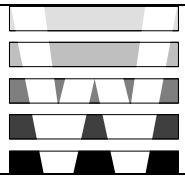
keine

3.7.17.3.3 Anforderungen an die Funktionen

LQS17.F.1 Nicht initialisierte Kalender
Es ist möglich, sich alle Ressourcen auflisten zu lassen, die einen nicht initialisierten Kalender besitzen.

LQS17.F.2 Nicht initialisierte Tarife
Es ist möglich, sich alle Ressourcen auflisten zu lassen, die nicht initialisierte Tarife besitzen.

LQS17.F.3 V-Modell-Rollen
Es ist möglich zu prüfen, ob alle im Projekthandbuch erwähnten Rollen des V-Modells Mitarbeitern zugeordnet wurden.



LQS17.F.4 Rollenzuordnung
Es ist möglich zu prüfen, ob allen Projektmitarbeitern eine Rolle des V-Modells zugeordnet wurde.

LQS17.F.5 Aktivitätenzuordnung
Es ist möglich zu prüfen, ob allen Projektmitarbeitern ein Arbeitsauftrag zugeordnet wurde.

LQS17.F.6 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.17.3.4 Sonstige Anforderungen

LQS17.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS17.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS17.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS17.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS17.So.5 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.18 Leistungseinheit LQS18 „Prüfungen der Projektorganisationspläne unterstützen“

3.7.18.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstand
„Projekthandbuch.*Projektorganisation*“)

Methode

keine

3.7.18.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen geprüft werden kann,

– ob alle im V-Modell aufgeführten Rollen auf betriebliche Organisationseinheiten abgebildet wurden und
– ob die Organisationseinheiten für die Abwicklung von Aktivitäten aus dem V-Modell benannt wurden.

3.7.18.3 Anforderungen

3.7.18.3.1 Anforderungen an die Schnittstellen

LQS18.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.18.3.2 Anforderungen an die Methodenunterstützung

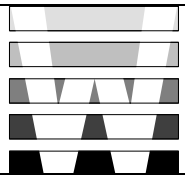
keine

3.7.18.3.3 Anforderungen an die Funktionen

LQS18.F.1 Zuordnung von Organisationseinheiten
Es ist möglich zu überprüfen, ob alle V-Modell-Rollen des Projekthandbuches betrieblichen Organisationseinheiten zugeordnet wurden.

Betriebliche Organisationseinheiten können sowohl Personengruppen als auch einzelne Personen sein.

LQS18.F.2 Rollenzuordnung
Es ist möglich zu prüfen, ob allen betrieblichen Organisationseinheiten Rollen des V-Modells zugeordnet wurden.



LQS18.F.3 Aktivitätenzuordnung
Es ist möglich zu prüfen, ob alle Aktivitäten des Projekthandbuches einer betrieblichen Organisationseinheit zugeordnet wurden.

LQS18.F.4 Rechtezuordnung

LQS18.F.4.1 Werkzeugfunktionen
Es ist möglich zu prüfen, ob allen betrieblichen Organisationseinheiten die nach V-Modell benötigten Rechte für die Benutzung von Werkzeugfunktionen zugeordnet wurden.

LQS18.F.4.2 Produkte
Es ist möglich zu prüfen, ob allen betrieblichen Organisationseinheiten die nach V-Modell benötigten Lese-/Schreibrechte für Produkte zugeordnet wurden.

LQS18.F.5 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung zu protokollieren.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.18.3.4 Sonstige Anforderungen

LQS18.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS18.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS18.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS18.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS18.So.5 Invarianz
Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.7.19 Leistungseinheit LQS19 „Prüfungen der Projektablaufpläne unterstützen“

3.7.19.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– QS 4.2 Produkt inhaltlich prüfen
(Prüfgegenstände
„Projektplan.*Projektablauf*“,
„Projektplan.*Aufwands- und Terminplan*“,
„Projektplan.*Einsatzmittelplan*“,
„Integrationsplan.*Organisation der Integration*“,
„Prüfplan.*Zeitplan*“ und
„Prüfplan.*Ressourcen*“)

Methode

keine

3.7.19.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen geprüft werden kann,

– ob alle projektspezifisch ausgewählten Aktivitätentypen konkret ausgeprägt wurden,
– ob alle Aktivitäten zeitlich zueinander in Beziehung gesetzt wurden,
– ob allen Aktivitäten Ressourcen und fixe Kosten zugeordnet wurden und
– ob diese Ressourcen höchstens bis zu ihrer Maximalkapazität und keinesfalls darüber hinaus verplant wurden.

3.7.19.3 Anforderungen

3.7.19.3.1 Anforderungen an die Schnittstellen

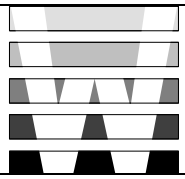
LQS19.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

3.7.19.3.2 Anforderungen an die Methodenunterstützung

keine

3.7.19.3.3 Anforderungen an die Funktionen

LQS19.F.1 Aktivitätenausprägung
Es ist möglich, zu überprüfen, ob alle projektspezifisch ausgewählten Aktivitätentypen konkret ausgeprägt wurden.



LQS19.F.2 Aktivitätenentsprechung
Es ist möglich, zu überprüfen, ob die ausgeprägten Aktivitäten den Aktivitätentypen des Projekthandbuches entsprechen.

Es dürfen keine Aktivitäten bezüglich gestrichener Aktivitätentypen existieren.

LQS19.F.3 Anfangsverzögerung
Es ist möglich, Anfangsverzögerungen von Aktivitäten auflisten zu lassen.

LQS19.F.4 Endeverzögerung
Es ist möglich, Endeverzögerungen von Aktivitäten auflisten zu lassen.

LQS19.F.5 Ressourcenkonflikte
Es ist möglich, Ressourcenkonflikte melden zu lassen.

Ressourcenkonflikte entstehen durch eine zeitliche Überlastung von Ressourcen (z. B. die Überplanung eines Mitarbeiters).

LQS19.F.6 Negative Puffer
Es ist möglich, Meldungen über negative Puffer zu erhalten.

Negative Puffer entstehen, wenn der Termin für das spätestmögliche Ende einer Aktivität vor dem Soll-Ende der Aktivität liegt.

LQS19.F.7 Zyklen im Netzplan
Es ist möglich, Meldungen über Zyklen im Netzplan zu erhalten.

Zyklische Abhängigkeiten zwischen den Aktivitäten eines Netzplans deuten auf Fehler in der Planung hin.

LQS19.F.8 Protokollierung
Es ist möglich, die Ergebnisse der Prüfung protokollieren zu lassen.

Mit „Protokollieren“ ist hier nur das Aufzeichnen aller Daten gemeint, die für eine spätere Dokumentation der Prüfung herangezogen werden können (z. B. Prüffälle, erwartete Ergebnisse, tatsächliche Ergebnisse, Meldungen), nicht jedoch die Aufbereitung der Dokumentation im Sinne von Prüfprotokollen selbst.

3.7.19.3.4 Sonstige Anforderungen

LQS19.So.1 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LQS19.So.2 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

LQS19.So.3 Hintergrund
Eine Prüfung läuft im Hintergrund ab; es sei denn, es wird eine Prüfung explizit im Vordergrund gewünscht.

Da eine Prüfung unter Umständen längere Zeit in Anspruch nehmen kann, muß beim Ablauf im Hintergrund nicht auf das Ende gewartet werden, sondern es können in der Zwischenzeit andere Arbeiten ausgeführt werden.

LQS19.So.4 Endesignal
Das Ende einer Prüfung wird signalisiert.

Dies kann akustisch (z. B. durch einen Piepton) oder optisch (z. B. durch ein spezielles Fenster mit einer entsprechenden Meldung) geschehen.

LQS19.So.5 Invarianz

Die Anwendung des Werkzeugs verändert den Prüfgegenstand nicht.

3.8 V-Modell: Leistungskomplex KM

3.8.1 Leistungseinheit LKM01 „Konfigurationsplanung und -kontrolle unterstützen“

3.8.1.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- KM 2.2 Konfiguration initialisieren („Konfigurations-Identifikationsdokument“)
- KM 2.4 Konfiguration fortschreiben („Konfigurations-Identifikationsdokument“)

Methode

keine

3.8.1.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen

- eine Konfigurationsstruktur geplant werden kann;
- Produkte und deren Versionen bestimmten Konfigurationen zugeordnet werden können;
- Produkte, die als Grundlage für die weiteren Entwicklungsarbeiten dienen sollen, den Baselines zugeordnet werden können;
- im Laufe der Systementwicklung die entsprechenden Produkte in die Konfigurationsstruktur aufgenommen werden können und
- ein Überblick über Konfigurationen ausgegeben werden kann.

3.8.1.3 Anforderungen

3.8.1.3.1 Anforderungen an die Schnittstellen

LKM01.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LKM01.S.2 Eingangs-Schnittstelle zu LKM02 „Versionen verwalten“
Es ist möglich, Versionen von Speicherobjekten ohne weitere Transformation aus LKM02 für die Einbindung in die Konfigurationsplanung und -kontrolle zu übernehmen.

LKM01.S.3 Ausgangs-Schnittstelle zu LOBV01 „Datenstrukturen und Daten der SEU erstellen, speichern, verwalten“
Es ist möglich, Informationen über Konfigurationsstrukturen und über Baselines in LOBV01 als Speicherobjekte abzulegen.

3.8.1.3.2 Anforderungen an die Methodenunterstützung

keine

3.8.1.3.3 Anforderungen an die Funktionen

LKM01.F.1 Darstellbarkeit von Konfigurationen

LKM01.F.1.1 Hierarchie
Es ist möglich, hierarchische Strukturen zu behandeln.

LKM01.F.1.2 Netzwerk
Es ist möglich, nicht-hierarchische (netzwerkartige) Strukturen zu behandeln.

LKM01.F.1.3 Dokumentstruktur
Komplexe Textdokumente sind durch strukturierte Teildokumente darstellbar.

Gewisse Textdokumente (z. B. Handbücher) sind Konfigurationsobjekte, die auf Grund ihrer Größe oder Inhomogenität sowohl bei der Erstellung als auch bei der späteren Verwaltung die Manipulation in kleineren Einheiten verlangen, nicht zuletzt auch, weil Teile davon in anderen Konfigurationen zusätzlich Verwendung finden können (Redundanzfreiheit).

LKM01.F.1.4 Beziehungen
Es ist möglich, Konfigurationsobjekte zu anderen Konfigurationsobjekten in Beziehung zu setzen.

Dies gilt für komplexe wie für elementare Konfigurationsobjekte, insbesondere auch für Verweise auf Konfigurationsobjekte, die nicht im Speichersystem enthalten sind, wie z. B. Hardware und Geräte.

LKM01.F.1.5 Variable Beziehungen
Es ist möglich, variable Beziehungen zwischen Objekten in einer Konfigurationsstruktur zu behandeln.

LKM01.F.1.6 Konfigurationsattribute
Es ist möglich, die Konfiguration und alle Konfigurationsobjekte mit verschiedenen, einzeln ansprechbaren Zusatzinformationen auszustatten.

Beispiel: Beschreibung der Merkmale/Besonderheiten einer Version, ihrer charakteristischen Unterschiede zu vorausgehenden Versionen oder zu funktionsgleichen Varianten; Hinweise über den Grund zur Versionsbildung; Bezug auf die Änderungsdokumente.

LKM01.F.1.7 Produktattribute
Die vom V-Modell geforderten Produktattribute sind darstellbar.

Die Attribute stellen zu speichernde Informationen dar, die nicht zwingend als „Attribut“ im physischen Sinne behandelt werden müssen.

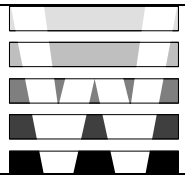
LKM01.F.1.8 Baselines
Es ist möglich, Baselines zu definieren.

Alle Produkte eines Projekts, die zu einer bestimmten Baseline gehören, werden zueinander in Beziehung gesetzt.

LKM01.F.1.9 Anzahl
Die Anzahl möglicher Konfigurationsobjekte und Versionen ist nicht begrenzt.

LKM01.F.2 Vorgaben und Kontrollen

LKM01.F.2.1 Konfigurationsstruktur
Die Definition der Struktur und der Strukturelemente einer Konfiguration vor deren Existenz ist möglich.



Es muß möglich sein, den Aufbau einer Konfiguration zu bestimmen und im Werkzeug zu etablieren, bevor die Versionen (Inhalte) konkret eingegeben werden.

LKM01.F.2.2 Änderungsberechtigung
Die Änderung einer Struktur wird durch Berechtigungen überwacht.

LKM01.F.2.3 Nachvollziehbarkeit
Die Änderung einer Struktur ist nachvollziehbar.

LKM01.F.2.4 Wertebereich
Es ist möglich, den Attributen einen Wertebereich zuzuweisen, dessen Einhaltung kontrolliert werden kann.

LKM01.F.2.5 KID-Gültigkeit
Die Gültigkeit des Konfigurations-Identifikationsdokuments (KID) wird kontrolliert.

Nachführungen des KID können automatisch oder manuell erfolgen. Bei manueller Nachführung können die Einträge gegen die vorhanden, akzeptierten Konfigurationsobjekte geprüft werden; umgekehrt kann beim Zustandswechsel eines Konfigurationsobjektes das KID kontrolliert werden.

LKM01.F.2.6 Generierung
Es ist möglich, ein Produkt aus seinen Teilen zu generieren.

Voraussetzung ist, daß Abspeicherung und Verwaltung von Dokumenten in Teilen erfolgt.

LKM01.F.2.7 Teilkonfiguration
Es ist möglich, Teile einer Konfiguration in eine andere Konfiguration zu übernehmen.

Generierungsmerkmale sind z. B. bestimmte Versionen oder Varianten, textliche Repräsentation, Weitergabe an andere Werkzeuge.

LKM01.F.2.8 Mehrfachverwendung
Es ist möglich, identische Konfigurationsobjekte in verschiedenen Konfigurationen zu verwenden.

Dies dient zur Reduzierung des Verwaltungs- und Wartungsaufwandes, da identische Konfigurationsobjekte in einer Konfigurationsfamilie nicht mehrfach gepflegt werden müssen.

LKM01.F.3 Unterschiede

LKM01.F.3.1 Strukturunterschiede
Es ist möglich, strukturelle Unterschiede von Konfigurationen darzustellen.

Die Ermittlung struktureller Unterschiede ist in jedem Falle durch das Werkzeug selbst zu leisten, da dieses allein im Besitz der dazu erforderlichen Daten ist.

LKM01.F.3.2 Attributunterschiede
Es ist möglich, die Unterschiede in den Attributen darzustellen (Beschreibung, Attribute, Zustandsdaten, Verwaltungsdaten).

Die Ermittlung der Unterschiede ist in jedem Falle durch das KM-Werkzeug selbst zu leisten, da dieses allein im Besitz der dazu erforderlichen Daten ist.

LKM01.F.4 Anforderungen an die Handhabung

LKM01.F.4.1 Navigation
Es ist eine Navigationsfunktion zum schnellen Auffinden von Informationen verfügbar.

Die aktuelle Position in der Struktur soll sichtbar gemacht werden, z. B. durch Darstellung des Wegs vom Einstieg bis zu Objekten unterhalb der aktuellen Version.

LKM01.F.4.2 Konfigurationsgraph
Es ist möglich, die Konfigurationsstruktur grafisch darzustellen.

LKM01.F.4.3 Konfigurationsübersicht
Es ist möglich, die gesamte Konfiguration in einer Übersicht darzustellen.

LKM01.F.4.4 Rechtesituation
Es ist möglich, die Rechtesituation zu einer Version einer Konfiguration insgesamt zu überblicken.

LKM01.F.4.5 Zustandsübersicht
Es ist möglich, die Zustände innerhalb einer Version einer Konfiguration insgesamt zu überblicken.

LKM01.F.4.6 Bibliotheken
Es ist möglich, Bibliotheken als Objekte zu verwalten.

Damit erfolgt die Manipulation der Bibliotheken unter Werkzeugkontrolle.

LKM01.F.4.7 Kontrolle der Auslieferung

LKM01.F.4.7.1 Auslieferungsversionen
Auslieferungen werden als Versionen verwaltet.

Eine Auslieferung verfügt über spezielle Begleitdaten, z. B. Adressat, Installation, Liefertermin, Release (Versionsnummer, Variante), Abnahmetermin, Bezüge zu Problemmeldungen, usw.

LKM01.F.4.7.2 Nachvollziehbarkeit
Jede Auslieferung ist nachvollziehbar.

Es muß kontrollierbar sein, welcher Anwender der Konfiguration eine Kopie einer Version erhalten hat; nicht zuletzt wegen der Verwaltung der Problemmeldungen und der notwendigen Auslieferung von neuen Releases (Revisionen, Versionen).

3.8.1.3.4 Sonstige Anforderungen

LKM01.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LKM01.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LKM01.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.8.2 Leistungseinheit LKM02 „Versionen verwalten“

3.8.2.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität

- KM 2.1 Produkt initialisieren
- KM 2.3 Produkt verwalten

Methode

keine

3.8.2.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen

- Versionen von Produkten definiert werden können und
- die Änderungshistorie von Produkten erstellt werden kann.

3.8.2.3 Anforderungen

3.8.2.3.1 Anforderungen an die Schnittstellen

LKM02.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LKM02.S.2 Ausgangs-Schnittstelle zu LKM01 „Konfigurationsplanung und -kontrolle unterstützen“
Es ist möglich, Versionen von Speicherobjekten ohne weitere Transformation aus LKM01 für die Einbindung in die Konfigurationsplanung und -kontrolle zu übernehmen.

LKM02.S.3 Ausgangs-Schnittstelle zu LOBV01 „Datenstrukturen und Daten der SEU erstellen, speichern, verwalten“
Es ist möglich, Informationen über Konfigurationsstrukturen und über Baselines in LOBV01 als Speicherobjekte abzulegen.

3.8.2.3.2 Anforderungen an die Methodenunterstützung

keine

3.8.2.3.3 Anforderungen an die Funktionen

LKM02.F.1 Verwaltung der Konfigurationsobjekte

LKM02.F.1.1 Attribute
Es ist möglich, Konfigurationsobjekte mit Attributen auszustatten.

LKM02.F.1.2 Attributeintrag
Die Werte folgender Attribute werden automatisch eingetragen und zur Kenntnis gebracht: Datum und Uhrzeit der Erstellung, Name des Bearbeiters und Name des Eigentümers.

LKM02.F.2 Identifikation der Konfigurationsobjekte

LKM02.F.2.1 Identifikation
Es ist möglich, Konfigurationsobjekte eindeutig zu identifizieren.

LKM02.F.2.2 Bezeichnung
Unterschiedliche Versionen sind an ihrer Bezeichnung erkennbar.

LKM02.F.3 Mischen von Versionen

LKM02.F.3.1 Mischen von Versionen
Es ist möglich, ein Konfigurationsobjekt aus Objekten mit verschiedenen Versionsständen zu mischen.

LKM02.F.3.2 Information über Konflikte
Eine durch Mischen entstandene Version enthält Informationen über aufgetretene Konflikte.

Konflikte treten dort auf, wo beide Ausgangsversionen sich so unterscheiden, daß die Mischung nicht durch eine einfache „Und"-Operation zu bewerkstelligen ist, sondern eine „Entweder/oder"-Entscheidung verlangt.

LKM02.F.4 Vergleichen von Versionen

LKM02.F.4.1 Vergleichen von Versionen
Es ist möglich, beliebige Versionen beliebiger Konfigurationsobjekte auf Unterschiede hin zu untersuchen.

LKM02.F.4.2 Anzahl/Ort der Unterschiede
Die Anzahl und der Ort der Unterschiede werden dargestellt.

LKM02.F.4.3 Inhalt der Unterschiede
Der Inhalt der Unterschiede wird dargestellt.

Die Umgebung der unterschiedlichen Zeilen wird angezeigt.

LKM02.F.4.4 Umgebungsgröße
Es ist möglich, die Größe der Umgebung zu wählen.

LKM02.F.5 Kontrolle des Entwicklungsstandes

LKM02.F.5.1 V-Modell-Zustände
Die vom V-Modell geforderten Zustände (Zustandswechsel) sind darstellbar.

Das V-Modell fordert folgende Produktzustände: geplant, in Bearb., vorgelegt und akzeptiert.

LKM02.F.5.2 Durchgängigkeit
Jedes Konfigurationsobjekt wird mit einem Zustand gemäß dem V-Modell versehen.

LKM02.F.5.3 Attribute für Zustände
Die Zustände verfügen über Attribute.

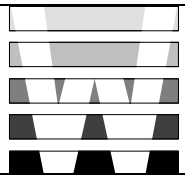
Beispielsweise wann erreicht, von wem vorgelegt, von wem akzeptiert.

LKM02.F.6 Nachvollziehbarkeit von Änderungen

LKM02.F.6.1 Historie
Es wird eine Änderungshistorie geführt.

LKM02.F.6.2 Detailangaben
Die Änderungshistorie enthält mindestens folgende Angaben: Datum und Uhrzeit der Änderung, Bearbeiter, Lokation jeder einzelnen Änderung, Kommentar.

Der Kommentar erläutert u. a. die Hintergründe, Absichten und Auswirkungen der Änderung.



LKM02.F.6.3 Darstellung
Jedes Konfigurationsobjekt ist so darstellbar, daß die einzelnen Änderungen dem Text zugeordnet sind.

Jede Zeile wird z. B. mit der Versionsnummer des entsprechenden Konfigurationsobjektes dargestellt.

LKM02.F.6.4 Information über neue Version
Jede zwischenzeitliche Erzeugung einer neuen Version seit der letzten Nutzung des Konfigurationsobjektes wird zur Kenntnis gebracht.

LKM02.F.7 Änderungsmanagement

LKM02.F.7.1 Zuordnung
Es ist möglich, alle Änderungsdokumente den entsprechenden Versionen zuzuordnen.

LKM02.F.7.2 Betroffene Versionen
Es ist möglich, alle von einer Änderung betroffenen Versionen mit den entsprechenden Änderungsdokumenten in bezug zu setzen.

LKM02.F.7.3 Änderungszustände
Mindestens alle im V-Modell geforderten Zustände für das Änderungswesen sind darstellbar.

LKM02.F.7.4 Darstellung des Änderungsgeschehens
Es ist möglich, das Änderungsgeschehen zu einer Version darzustellen.

Beispielsweise: welche Teile einer Konfiguration sind zur Änderung freigegeben, welche Teile wurden tatsächlich geändert, wann wurde die Änderung ausgeführt, in welchem Zustand befindet sich der Änderungsvorgang, von wem wurde die Änderung vorgenommen, von wem wurde die Änderung abgenommen.

LKM02.F.7.5 Logbuch
Änderungen werden in entsprechenden Konfigurationsobjekten eingetragen, so daß die Änderungshistorie bei der Nutzung des Konfigurationsobjektes jederzeit sichtbar ist.

LKM02.F.7.6 Statistik
Es ist möglich, Übersichten und Auswertungen über die aktuelle und vergangene Situation der Änderungsvorgänge (Änderungsstatistik) auf Abruf zu generieren.

LKM02.F.7.7 Neue Version
Jede Änderung einer Version führt zu einer neuen Version.

LKM02.F.7.8 Folgeänderungen
Es ist möglich, auf der Basis der Konfigurationsstruktur die von der Änderung eines Objektes in Folge betroffenen Objekte aufzuzeigen. Dabei können die Objekte sowohl über hierarchische, netzwerkartige als auch variable Strukturen in Beziehung gesetzt sein.

LKM02.F.8 Handhabung

LKM02.F.8.1 Navigation
Es ist eine Navigationsfunktion zum schnellen Auffinden von Informationen verfügbar.

Die aktuelle Position in der Struktur soll sichtbar gemacht werden, z. B. durch Darstellung des Wegs zum Einstieg bis zu Objekten unterhalb der aktuellen Version.

LKM02.F.8.2 Mehrere Versionen
Es ist möglich, mehrere Versionen gleichzeitig zu bearbeiten.

LKM02.F.8.3 Zustandsübersicht
Es ist möglich, die Zustände innerhalb einer Version einer Konfiguration insgesamt zu überblicken.

LKM02.F.9 Struktur eines Konfigurationsobjektes
Es ist möglich, die Struktur eines Konfigurationsobjektes vor dessen Entstehen zu definieren.

LKM02.F.10 Kontrolle der Attribute
Die Verwendung der Attribute wird kontrolliert.

Die Eintragung von Attributen erfolgt automatisch oder wird vom Nutzer erzwungen.

LKM02.F.11 Kontrolle des KID
Die Gültigkeit des Konfigurations-Identifikationsdokuments (KID) wird kontrolliert.

Nachführungen des KID können automatisch oder manuell erfolgen. Bei manueller Nachführung können die Einträge gegen die vorhandenen, akzeptierten Konfigurationsobjekte geprüft werden; umgekehrt kann beim Zustandswechsel eines Konfigurationsobjektes das KID kontrolliert werden.

3.8.2.3.4 Sonstige Anforderungen

LKM02.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LKM02.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LKM02.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.9 V-Modell: Leistungskomplex PM

3.9.1 Leistungseinheit LPM01 „Projektressourcen verwalten“

3.9.1.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– PM 1.5 Grobplan erstellen („Projektplan.*Einsatzmittelplan*“)

– PM 4 Feinplanung („Projektplan.*Einsatzmittelplan*“)

– PM 11 Bereitstellung der Ressourcen

Methode

keine

3.9.1.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen projektspezifisch

– die verfügbaren Ressourcen verwaltet sowie
– den Mitarbeitern die Rollen gemäß V-Modell zugeordnet werden.

Bei der Verwaltung verfügbarer Ressourcen werden u. a. Informationen zu Ressourcen initialisiert, bei Bedarf geändert und wieder gelöscht.

3.9.1.3 Anforderungen

3.9.1.3.1 Anforderungen an die Schnittstellen

LPM01.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LPM01.S.2 Ausgangs-Schnittstelle zu LPM02 „Planung der Projektorganisation unterstützen“
Es ist möglich, Informationen zu Ressourcen ohne weitere Transformation über die Objektverwaltung an LPM02 zur Erstellung von Projektorganisationsplänen zu übergeben.

LPM01.S.3 Ausgangs-Schnittstelle zu LPM04 „Planung des Projektablaufs unterstützen“
Es ist möglich, Informationen zu Ressourcen ohne weitere Transformation über die Objektverwaltung an LPM04 zur Erstellung von Projektablaufplänen zu übergeben.

LPM01.S.4 Ausgangs-Schnittstelle zu LPM05 „Kostenschätzung unterstützen“
Es ist möglich, Informationen zu Ressourcengruppen ohne weitere Transformation über die Objektverwaltung an LPM05 zur Verwendung bei der Kostenschätzung zu übergeben.

3.9.1.3.2 Anforderungen an die Methodenunterstützung

keine

3.9.1.3.3 Anforderungen an die Funktionen

LPM01.F.1 Ressourceneingaben
Es ist möglich, Ressourcen zu erfassen.

LPM01.F.2 Ressourcengruppen
Es ist möglich, Ressourcengruppen zu erfassen bzw. vorhandene Ressourcen zu Gruppen zusammenzufassen.

LPM01.F.3 Informationen zu Ressourcen

LPM01.F.3.1 Freitext
Es ist möglich, für Ressourcen Freitext-Beschreibungen zu erfassen.

LPM01.F.3.2 Identifikatoren
Es ist möglich, Ressourcen mit Identifikatoren zu versehen (eindeutige Namen).

LPM01.F.3.3 Attribute
Es ist möglich, für Auswertungen Ressourcen mit Attributen zu versehen.

LPM01.F.3.4 Gruppierung
Es ist möglich, Ressourcen Gruppen zuzuordnen.

LPM01.F.3.5 V-Modell-Rollen
Es ist möglich, den Ressourcen die Rollen des V-Modells zuzuordnen.
Ressourcen sind hier die Projektmitarbeiter.

LPM01.F.4 Ressourcenklassifikation

LPM01.F.4.1 Nichtverbrauchbare Ressourcen
Es ist möglich, Ressourcen als „nichtverbrauchbar“ zu klassifizieren.
Zu den nichtverbrauchbaren Ressourcen zählt beispielsweise das Personal.

LPM01.F.4.2 Verbrauchbare Ressourcen
Es ist möglich, Ressourcen als „verbrauchbar“ zu klassifizieren.
Zu den verbrauchbaren Ressourcen zählt beispielsweise Material.

LPM01.F.5 Ressourcentypen

LPM01.F.5.1 Ressourcentyp Personal
Es ist möglich, Ressourcen vom Typ Personal zu definieren.

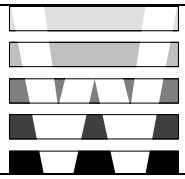
LPM01.F.5.2 Ressourcentyp Hardware/Software
Es ist möglich, Ressourcen vom Typ Hardware/Software zu definieren.

LPM01.F.5.3 Ressourcentyp Material
Es ist möglich, Ressourcen vom Typ Material zu definieren.

LPM01.F.6 Ressourcenzuordnung
Es ist möglich, Ressourcen den Aktivitäten des Projekthandbuchs zuzuordnen.
Dies geschieht durch Erteilung eines Arbeitsauftrages.

LPM01.F.7 Maßeinheiten

LPM01.F.7.1 Zeitbezogen
Es ist möglich, zeitbezogene Maßeinheiten zu definieren.



Zu den zeitbezogenen Maßeinheiten zählen beispielsweise: Jahr, Monat, Woche, Kalendertag, Arbeitstag, Stunden, Minuten.

LPM01.F.7.2 Mengenbezogen
Es ist möglich, mengenbezogene Maßeinheiten zu definieren.

Für den Ressourcentyp Material muß die Maßeinheit Anzahl oder Stück vergeben werden.

LPM01.F.7.3 Wertbezogen
Es ist möglich, für den Ressourcenverbrauch Tarife (Kosten) zu definieren.

LPM01.F.8 Kalender

LPM01.F.8.1 Kalendereingabe
Es ist möglich, für Projekte und Mitarbeiter individuelle Kalender zu erfassen.

LPM01.F.8.2 Verfügbarkeit
Es ist möglich, die Verfügbarkeit von Ressourcen über die Zeit linear oder nicht-linear zu definieren.

LPM01.F.8.3 Hierarchische Kalender
Die Kalender für Ressourcen und Projekte stehen in einer hierarchischen Beziehung zueinander.

Die Kalendertermine (z. B. Feiertage, arbeitsfreie Tage bezogen z. B. auf Land, Firma, Abteilung, Mitarbeiter) werden in der Hierarchie vererbt.

LPM01.F.8.4 Priorisierung
Es ist möglich, für Kalender Prioritäten zu vergeben.

Die Hierarchie der Kalender kann durch Priorisierung überschrieben werden.

LPM01.F.9 Datumsinformationen
Die Eingabe nicht gültiger Datumsinformationen ist nicht möglich.

3.9.1.3.4 Sonstige Anforderungen

LPM01.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LPM01.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LPM01.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.9.2 Leistungseinheit LPM02 „Planung der Projektorganisation unterstützen“

3.9.2.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– PM 1.1 Projekt einrichten
(„Projekthandbuch.*Projektorganisation*“)

Methode

– Organigramm (OGG)

3.9.2.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen projektspezifisch

– die im V-Modell aufgeführten Rollen auf betriebliche Organisationseinheiten abgebildet und
– die Instanzen benannt werden, die für die Abwicklung von Aktivitäten aus dem V-Modell zuständig sind.

3.9.2.3 Anforderungen

3.9.2.3.1 Anforderungen an die Schnittstellen

LPM02.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LPM02.S.1 Eingangs-Schnittstelle zu LPM01 „Projektressourcen verwalten“
Es ist möglich, mit LPM01 erfaßte Informationen zu Ressourcen ohne weitere Transformation aus der Objektverwaltung zur Erstellung von Projektorganisationsplänen zu übernehmen.

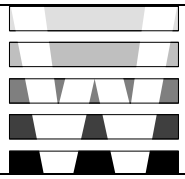
3.9.2.3.2 Anforderungen an die Methodenunterstützung

LPM02.M.1 Organigramm (OGG)

LPM02.M.1.1 Mikro-Organisation

LPM02.M.1.1.1 Kontrollierte, zentralisierte Organisation
Die Darstellung einer kontrollierten, zentralisierten Organisationsform eines Projektes ist möglich (Chief Programmer Team).

LPM02.M.1.1.2 Demokratisch, dezentralisierte Organisation
Die Darstellung einer demokratisch, dezentralisierten Organisationsform eines Projektes ist möglich (Egoless Programmer Team).



LPM02.M.1.2 Makro-Organisation

LPM02.M.1.2.1 Projektorganisation
Die Darstellung einer Projektorganisation ist möglich.

LPM02.M.1.2.2 Stab-/Linienorganisation
Die Darstellung einer Stab-/Linienorganisation ist möglich.

LPM02.M.1.2.3 Matrixorganisation
Die Darstellung einer Matrixorganisation ist möglich.

3.9.2.3.3 Anforderungen an die Funktionen

LPM02.F.1 Rollenzuordnung
Es ist möglich, den betrieblichen Organisationseinheiten die Rollen des V-Modells zuzuordnen.

LPM02.F.2 Mehrfachrollen
Eine betriebliche Organisationseinheit kann mehrere Rollen übernehmen; dabei werden miteinander unvereinbare Rollen zurückgewiesen.

LPM02.F.3 Aktivitätentypen
Es ist möglich, den betrieblichen Organisationseinheiten entsprechend den Rollen des V-Modells Aktivitätentypen zuzuweisen.

3.9.2.3.4 Sonstige Anforderungen

LPM02.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LPM02.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LPM02.So.3 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.9.3 Leistungseinheit LPM03 „Ausschreibungsrelevantes Tailoring unterstützen“

3.9.3.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– PM 1.3 Projektspezifisches V-Modell erstellen („Projekthandbuch.*Projektspezifisches V-Modell*“)

Methode

keine

3.9.3.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen projektspezifisch

– Haupt- oder Teilaktivitätentypen aus dem allgemeinen V-Modell herausgestrichen und
– Streichbedingungen für das Technische Tailoring identifiziert und beschrieben

werden.

Nach Beantwortung von Fragen zu den Ausschreibungsrelevanten Streichbedingungen wird ein projektspezifisches V-Modell erzeugt. Diese Funktion wird „geführtes Tailoring“ genannt.

Aktivitätentypen der inneren Struktur des V-Modells können unabhängig von Streichbedingungen gestrichen werden. Diese Funktion wird „ungeführtes Tailoring“ genannt.

3.9.3.3 Anforderungen

3.9.3.3.1 Anforderungen an die Schnittstellen

LPM03.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LPM03.S.2 Ausgangs-Schnittstelle zu LSES01 „Systementwicklung steuern“
Es ist möglich, projektspezifische V-Modelle ohne weitere Transformation über die Objektverwaltung an LSES01 zur Steuerung der Systementwicklung zu übergeben.

LPM03.S.3 Ausgangs-Schnittstelle zu LPM04 „Planung des Projektablaufs unterstützen“
Es ist möglich, projektspezifische V-Modelle ohne weitere Transformation über die Objektverwaltung an LPM04 zu Erstellung von Projektablaufplänen zu übergeben.

LPM03.S.4 Ausgangs-Schnittstelle zu LPM05 „Kostenschätzung unterstützen“
Es ist möglich, projektspezifische V-Modelle ohne weitere Transformation über die Objektverwaltung an LPM05 zur Verwendung bei der Kostenschätzung zu übergeben.

3.9.3.3.2 Anforderungen an die Methodenunterstützung

keine

3.9.3.3.3 Anforderungen an die Funktionen

LPM03.F.1 Abbilden des V-Modells

LPM03.F.1.1 Laden des V-Modells
Es ist möglich, die inneren Struktur des V-Modells zu laden.

Die innere Struktur des V-Modells enthält Aktivitätentypen, Produkttypen und Beziehungen zwischen Aktivitätentypen und Produkttypen,

LPM03.F.1.2 Ausschreibungsrelevante Streichbedingungen
Es ist möglich, die ausschreibungsrelevanten Streichbedingungen zu laden.

LPM03.F.1.3 Technische Streichbedingungen
Es ist möglich, technische Streichbedingungen zu laden.

LPM03.F.1.4 Laden des V-Modelltextes
Es ist möglich, die Texte des V-Modells, insbesondere die Abwicklungstexte und die Produktbeschreibungen zu laden.

LPM03.F.2 Erzeugen eines projektspezifischen V-Modells

LPM03.F.2.1 Sequentiell
Streichbedingungen werden sequentiell am Bildschirm vom Nutzer beantwortet.

LPM03.F.2.2 Neubeantwortung
Es ist möglich, zu bereits beantworteten Fragestellungen zurückzugehen und diese neu zu beantworten.

LPM03.F.2.3 Keine überflüssigen Fragen
Fragestellungen, die aufgrund einer vorhergehenden Antwort indirekt bereits beantwortet worden sind, werden nicht mehr angezeigt.

LPM03.F.2.4 Tailoring-Vorschlag
Nach dem Durchlaufen von allen in Frage kommenden Projektkriterien wird ein Tailoring-Vorschlag erstellt. Darin sind je nach Beantwortung der Streichbedingungen bereits bestimmte Aktivitätentypen und evtl. Produkttypen der inneren Struktur gestrichen (oder Produktkapitel als „nicht relevant“ gekennzeichnet).

LPM03.F.3 Streichen von Aktivitätentypen

LPM03.F.3.1 Technische Streichbedingungen
Es ist möglich, Technische Streichbedingungen zu formulieren, an die die Durchführung einer Aktivität gebunden ist.

LPM03.F.3.2 Streichen von Produkttypen
Bei der Streichung eines Aktivitätentyps wird der aus ihm entstehende Produkttyp automatisch gestrichen oder das betroffene Produktkapitel als „nicht relevant“ gekennzeichnet.

Es können mehrere Produkttypen bzw. Produktkapitel von der Streichung betroffen sein.

LPM03.F.3.3 Teilaktivitätentyp streichen
Wurden alle Teilaktivitätentypen eines (Teil-) Aktivitätentyps gestrichen, so wird auch der (Teil-) Aktivitätentyp gestrichen.

LPM03.F.3.4 Teilaktivitätentyp automatisch streichen
Wird ein Aktivitätentyp gestrichen, so werden auch alle seine (beliebig geschachtelten) Teilaktivitätentypen gestrichen.

LPM03.F.3.5 Produkttyp streichen
Wird ein Aktivitätentyp gestrichen, so wird der erzeugte Produkttyp nur dann gestrichen, wenn dieser Aktivitätentyp das Produkt vollständig erzeugt.

LPM03.F.3.6 Explizites Streichen von Aktivitätentypen

Es ist möglich, Aktivitätentypen unabhängig von Streichbedingungen zu streichen.

LPM03.F.4 Streichen von Produkttypen

LPM03.F.4.1 Produkttyp streichen
Wurden alle Teilprodukttypen eines (Teil-) Produkttyps gestrichen, so wird auch der (Teil-) Produkttyp gestrichen.

LPM03.F.4.2 Teilprodukttyp automatisch streichen
Wird ein Produkttyp gestrichen, so werden auch alle seine (beliebig geschachtelten) Teilprodukttypen gestrichen.

LPM03.F.5 Abändern von Abwicklungstexten

LPM03.F.5.1 Streichungen
Es ist möglich, nicht zutreffende Passagen des Abwicklungstextes zu streichen.

LPM03.F.5.2 Anpassungen
Es ist möglich, sprachliche Anpassungen im Abwicklungstext vorzunehmen.

LPM03.F.5.3 Anwendungshinweise
Es ist möglich, den Abwicklungstext um für den Projektverlauf wichtige Anwendungshinweise zu ergänzen.

LPM03.F.5.4 Originaltext

LPM03.F.5.4.1 Abänderung von Passagen
Die Abwicklungstexte des allgemeinen V-Modells bestehen aus Teilen, die verändert werden dürfen, und aus solchen, die nicht abgeändert werden dürfen.

Wenn versucht wird, eine nicht abänderbare Passage im Abwicklungstext zu modifizieren, erscheint eine entsprechende Meldung, und die Änderung wird nicht durchgeführt.

LPM03.F.5.4.2 Kennzeichnung änderbarer Passagen
Diejenigen Textpassagen, die verändert werden dürfen, sind kenntlich gemacht.

LPM03.F.5.5 Kennzeichnung abgeänderter Passagen
Abgeänderte Textpassagen von Abwicklungstexten werden automatisch optisch gekennzeichnet.

Eine solche Kennzeichnung kann beispielsweise durch Verwendung eines anderen Fonts geschehen.

LPM03.F.6 Abspeichern und Verwalten des V-Modells

LPM03.F.6.1 Entwicklungsdatenbank
Das allgemeine V-Modell wird in der Entwicklungsdatenbank abgespeichert.

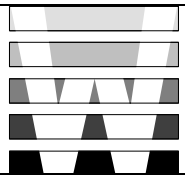
LPM03.F.6.2 Mehrere V-Modelle
Es ist möglich, gleichzeitig mehrere projektspezifische V-Modelle zu verwalten.

Damit können mehrere Projekte zeitgleich abgewickelt werden.

3.9.3.3.4 Sonstige Anforderungen

LPM03.So.1 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LPM03.So.2 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.



LPM03.So.3 Komplexität

Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.9.4 Leistungseinheit LPM04 „Planung des Projektablaufs unterstützen“

3.9.4.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- SE 2.6 System-Integration spezifizieren („System-Integrationsplan.*Organisation der Integration*“)
- SE 4.3-SW SW-Integration spezifizieren („SW-Integrationsplan.*Organisation der Integration*“)
- QS 1.2 Prüfplan erstellen („Prüfplan.*Zeitplan*“, „Prüfplan.*Ressourcen*“)
- PM 1.5 Grobplan erstellen („Projektplan.*Projektablauf*“, „Projektplan.*Aufwands- und Terminplan*“ und „Projektplan.*Einsatzmittelplan*“)
- PM 3 Auftragnehmer-Management („Berichtsdokumente“)
- PM 4 Feinplanung („Projektplan.*Projektablauf*“, „Projektplan.*Aufwands- und Terminplan*“ und „Projektplan.*Einsatzmittelplan*“)
- PM 7 Risikomanagement („Berichtsdokumente“)
- PM 8 Projektkontrolle und -steuerung („Berichtsdokumente“)

Methode

- Balkenplan (BALK)
- Netzplan-Technik (NPT)

3.9.4.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen

- projektspezifisch die ausgewählten Aktivitätentypen konkret ausgeprägt (Technisches Tailoring) werden,
- für Aktivitäten eine Terminplanung durchgeführt wird,
- die Aktivitäten zeitlich zueinander in Beziehung gesetzt werden,
- Meilensteine geplant und
- den Aktivitäten Ressourcen zugeordnet werden.

3.9.4.3 Anforderungen

3.9.4.3.1 Anforderungen an die Schnittstellen

LPM04.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LPM04.S.2 Eingangs-Schnittstelle zu LPM01 „Projektressourcen verwalten“
Es ist möglich, mit LPM01 erfaßte Informationen zu Ressourcen ohne weitere Transformation gezielt aus der Objektverwaltung zur Erstellung von Projektablaufplänen zu übernehmen.

LPM04.S.3 Eingangs-Schnittstelle zu LPM03 „Ausschreibungsrelevantes Tailoring unterstützen“
Es ist möglich, mit LPM03 erstellte projektspezifische V-Modelle ohne weitere Transformation gezielt aus der Objektverwaltung zur Erstellung von Projektablaufplänen zu übernehmen.

LPM04.S.4 Ausgangs-Schnittstelle zu LSES01 „Systementwicklung steuern“
Es ist möglich, Ausprägungen von Aktivitätentypen und Ressourcenzuordnungen aus Projektablaufplänen ohne weitere Transformation über die Objektverwaltung an LSES01 zur Steuerung der Systementwicklung zu übergeben.

LPM04.S.5 Ausgangs-Schnittstelle zu LPM05 „Kostenschätzung unterstützen“
Es ist möglich, Ausprägungen von Aktivitätenypen und Ressourcenzuordnungen aus Projektablaufplänen ohne weitere Transformation an LPM05 zur Schätzung der voraussichtlichen Aufwände zu übergeben.

LPM04.S.6 Ausgangs-Schnittstelle zu LPM06 „Projektkontrolle und -steuerung unterstützen“
Es ist möglich, Projektablaufpläne ohne weitere Transformation über die Objektverwaltung an LPM06 für die Projektkontrolle und -steuerung zu übergeben.

3.9.4.3.2 Anforderungen an die Methodenunterstützung

LPM04.M.1 Balkenplan (BALK)

LPM04.M.1.1 Symbole
Es ist möglich, Beziehungen zwischen Aktivitäten in Form eines Balkendiagrammes grafisch darzustellen.

LPM04.M.1.2 Meilensteine
Es ist möglich, Meilensteine festzulegen.

LPM04.M.2 Netzplan-Technik (NPT)

LPM04.M.2.1 Symbole
Es stehen unterschiedliche Symbole zur Darstellung von Aktivitäten und deren Beziehungen untereinander zur Verfügung.

LPM04.M.2.2 Ereignisknoten-Netz

LPM04.M.2.2.1 Optimistische Aktivitätendauer
Es ist möglich, in einem Ereignisknoten-Netz Aktivitäten eine optimistisch geschätzte Dauer zuzuordnen.

LPM04.M.2.2.2 Wahrscheinliche Aktivitätendauer
Es ist möglich, in einem Ereignisknoten-Netz Aktivitäten die wahrscheinliche Dauer zuzuordnen.

LPM04.M.2.2.3 Pessimistische Aktivitätendauer
Es ist möglich, in einem Ereignisknoten-Netz Aktivitäten eine pessimistisch geschätzte Dauer zuzuordnen.

LPM04.M.2.3 Vorgangsknoten-Netz
Es ist möglich, einen Netzplan in Form eines Vorgangsknoten-Netzes darzustellen (MPM).

LPM04.M.2.4 Vorgangspfeil-Netz
Es ist möglich, einen Netzplan in Form eines Vorgangspfeil-Netzes darzustellen (CPM).

LPM04.M.2.5 Kritischer Pfad
Es ist möglich, bei der grafischen Darstellung eines Netzplanes den kritischen Pfad optisch hervorzuheben.

LPM04.M.2.6 Kritische Aktivitäten
Es ist möglich, bei der grafischen Darstellung eines Netzplanes kritische Aktivitäten optisch hervorzuheben.

LPM04.M.2.7 Neuberechnung
Die Neuberechnung des kritischen Pfades in einem Netzplan ist konfigurierbar.

Die Berechnung wird manuell angestoßen oder erfolgt automatisch.

LPM04.M.2.8 Teil-Netze
Es ist möglich, Teil-Netze separat zu berechnen.

LPM04.M.2.9 Hierarchische Verfeinerung
Es ist möglich, Netzpläne hierarchisch zu verfeinern.

LPM04.M.2.10 Anordnungsbeziehungen

LPM04.M.2.10.1 Ende – Ende
Es ist möglich, zwischen zwei Aktivitäten eine Ende-Ende-Beziehung zu definieren.

LPM04.M.2.10.2 Ende – Start
Es ist möglich, zwischen zwei Aktivitäten eine Ende-Start-Beziehung zu definieren.

LPM04.M.2.10.3 Start – Ende
Es ist möglich, zwischen zwei Aktivitäten eine Start-Ende-Beziehung zu definieren.

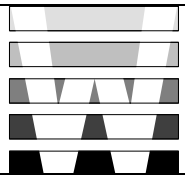
LPM04.M.2.10.4 Start – Start
Es ist möglich, zwischen zwei Aktivitäten eine Start-Start-Beziehung zu definieren.

LPM04.M.2.11 Ressourcenzuordnung
Es ist möglich, einer Aktivität Ressourcen zuzuordnen.

Dies geschieht durch Erteilung eines Arbeitsauftrages.

LPM04.M.2.12 Aktivitätendauer
Es ist möglich, für eine Aktivität den Starttermin und die Dauer festzulegen.

LPM04.M.2.13 Zeitlicher Bezug
Es ist möglich, Aktivitäten zeitlich zueinander in Beziehung zu setzen.



LPM04.M.2.14 Differenzzeiten
Es ist möglich, zwischen zwei Aktivitäten eine positive/negative Differenzzeit festzulegen.

LPM04.M.2.15 Frühester Start
Es ist möglich, für eine Aktivität den frühesten Starttermin festzulegen.

LPM04.M.2.16 Frühestes Ende
Es ist möglich, für eine Aktivität den frühesten Endtermin festzulegen.

LPM04.M.2.17 Spätester Start
Es ist möglich, für eine Aktivität den spätesten Starttermin festzulegen.

LPM04.M.2.18 Spätestes Ende
Es ist möglich, für eine Aktivität den spätesten Endtermin festzulegen.

LPM04.M.2.19 Fixtermine
Es ist möglich, für eine Aktivität Fixtermine festzulegen.

LPM04.M.2.20 Pufferzeiten
Es ist möglich, für eine Aktivität die Pufferzeit anzeigen zu lassen.

LPM04.M.2.21 Rückwärtsrechnung
Es ist möglich, auf der Basis aller spätestmöglichen Termine den spätesten Start und das späteste Ende jeder Aktivität zu berechnen.

Die Rückwärtsrechnung ermittelt den spätestmöglichen Termin für den Beginn eines Projektes.

LPM04.M.2.22 Vorwärtsrechnung
Es ist möglich, auf der Basis aller frühestmöglichen Termine den frühesten Start und das früheste Ende jeder Aktivität zu berechnen.

Die Vorwärtsrechnung ermittelt den kürzesten Projektverlauf und den frühestmöglichen Starttermin eines Projektes.

LPM04.M.2.23 Freitext
Es ist möglich, für Aktivitätentypen Freitext-Beschreibungen zu erfassen.

LPM04.M.2.24 Aufwand
Es ist möglich, für Aktivitätentypen bereits verbrauchten Aufwand zu erfassen.

LPM04.M.2.25 Teilpläne
Es ist möglich, Teilpläne zu einem Gesamtplan zu aggregieren und umgekehrt.

3.9.4.3.3 Anforderungen an die Funktionen

LPM04.F.1 Technisches Tailoring
Es ist möglich, die projektspezifisch ausgewählten Aktivitätentypen konkret auszuprägen.

LPM04.F.2 Streichbedingungen
Die im Projekthandbuch festgelegten projektspezifischen Streichbedingungen werden beim Ablauf des Technischen Tailoring berücksichtigt.

LPM04.F.3 Sammelaktivität
Es ist möglich, einzelne Aktivitäten zu Sammelaktivitäten zusammenzufassen. Diese werden als Einheit aufgefaßt.

LPM04.F.4 Ressourcenverbrauch pro Aktivität
Es ist möglich, den Ressourcenvcerbrauch einzelner Aktivitäten zu bestimmen.

LPM04.F.5 Prozentuale Fertigstellung
Es ist möglich, prozentuale Fertigstellungen einer Aktivität zu erfassen.

3.9.4.3.4 Sonstige Anforderungen

LPM04.So.1 Erfahrungsdatenbank
Es ist möglich, Plandaten in einer Erfahrungdatenbank abzulegen.

LPM04.So.2 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LPM04.So.3 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LPM04.So.4 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.9.5 Leistungseinheit LPM05 „Kostenschätzung unterstützen“

3.9.5.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

– PM 1.5 Grobplan erstellen
(„Projektplan.*Aufwands- und Terminplan*“)

– PM 4 Feinplanung
(„Projektplan.*Aufwands- und Terminplan*“)

Methode

– Schätzmodelle (SMOD)

3.9.5.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen die Prognose der im Projekt anfallenden Aufwände unterstützt wird. Dies geschieht auf der Grundlage von bereits vorhandenen Erfahrungswerten aus früheren Projekten, projektspezifischen Randbedingungen und unter der Annahme von Zukunftsentwicklungen.

3.9.5.3 Anforderungen

3.9.5.3.1 Anforderungen an die Schnittstellen

LPM05.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LPM05.S.2 Eingangs-Schnittstelle zu LPM01 „Projektressourcen verwalten“
Es ist möglich, mit LPM01 erfaßte Informationen zu Ressourcengruppen ohne weitere Transformation gezielt aus der Objektverwaltung für die Kostenschätzung zu übernehmen.

LPM05.S.3 Eingangs-Schnittstelle zu LPM03 „Ausschreibungsrelevantes Tailoring unterstützen“
Es ist möglich, mit LPM03 erstellte projektspezifische V-Modelle ohne weitere Transformation gezielt aus der Objektverwaltung für die Kostenschätzung zu übernehmen.

LPM05.S.4 Eingangs-Schnittstelle zu LPM04 „Planung des Projektablaufs unterstützen“
Es ist möglich, in LPM04 erfaßte Ausprägungen von Aktivitätentypen und Ressourcenzuordnungen aus Projektablaufplänen für die Kostenschätzung zu übernehmen.

LPM05.S.5 Ausgangs-Schnittstelle zu LPM06 „Projektkontrolle und -steuerung unterstützen“
Es ist möglich, Kostenschätzungen ohne weitere Transformation über die Objektverwaltung an LPM06 für die Projektkontrolle und -steuerung zu übergeben.

3.9.5.3.2 Anforderungen an die Methodenunterstützung

LPM05.M.1 Schätzmodelle (SMOD)

LPM05.M.1.1 Kosten

LPM05.M.1.1.1 Fixkosten
Es ist möglich, für Ressourcen Fixkosten zu erfassen.

Mit „Kosten" werden alle in Geldeinheiten umrechenbaren Aufwände und Ressourcenverbräuche bezeichnet.

LPM05.M.1.1.2 Variable Kosten
Es ist möglich, für Ressourcen variable Kosten zu erfassen.

LPM05.M.1.2 Akkumulierung von Einsatzmittelkosten

LPM05.M.1.2.1 Aktivitätentypen
Es ist möglich, die Einsatzmittelkosten bezogen auf Aktivitätentypen zu ermitteln.

LPM05.M.1.2.2 Ressourcengruppen
Es ist möglich, die Einsatzmittelkosten bezogen auf Ressourcengruppen zu ermitteln.

LPM05.M.1.2.3 Unterprojekte
Es ist möglich, die Einsatzmittelkosten bezogen auf Unterprojekte zu ermitteln.

LPM05.M.1.2.4 Zeitfenster
Es ist möglich, die Einsatzmittelkosten für ausgewählte Zeiträume zu ermitteln.

3.9.5.3.3 Anforderungen an die Funktionen

LPM05.F.1 Taschenrechner
Es ist möglich, Taschenrechnerfunktionen aufzurufen.

3.9.5.3.4 Sonstige Anforderungen

LPM05.So.1 Erfahrungsdatenbank
Es steht für Aufwandsschätzungen eine Erfahrungsdatenbank zur Verfügung,

In der Erfahrungsdatenbank werden Projektkennziffern bereits abgeschlossener Projekte abgespeichert, die für spätere Schätzungen herangezogen werden können.

LPM05.So.2 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LPM05.So.3 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LPM05.So.4 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.

3.9.6 Leistungseinheit LPM06 „Projektkontrolle und -steuerung unterstützen“

3.9.6.1 Zuordnung zum V-Modell und zur Methodenzuordnung

V-Modell-Aktivität (-Produkt)

- PM 8 Projektkontrolle und -steuerung („Berichtsdokumente“)
- KM 4.7 Projekthistorie führen („Projekthistorie.*Planungsstatistik*“)

Methode

- Earned Value Verfahren (EVV)
- Trend-Analyse (TRDA)

3.9.6.2 Kurzcharakteristik

In dieser Leistungseinheit werden Anforderungen an Werkzeuge gestellt, mit denen man Plandaten und Ist-Daten hinsichtlich Kosten, Terminen und Kapazitätsauslastung miteinander vergleichen kann. Dies kann sowohl auf explizite Anforderung hin als auch termingesteuert und ereignisgesteuert geschehen.

3.9.6.3 Anforderungen

3.9.6.3.1 Anforderungen an die Schnittstellen

LPM06.S.1 Granularität
Der Austausch von Steuerungsparametern mit LSES01 „Systementwicklung steuern“ ist für einzelne, in sich abgeschlossene Funktionspakete des Werkzeugs über eine offengelegte, dokumentierte Schnittstelle möglich.

LPM06.S.2 Eingangs-Schnittstelle zu LPM04 „Planung des Projektablaufs unterstützen“
Es ist möglich, mit LPM04 erfaßte Projektablaufpläne ohne weitere Transformation gezielt aus der Objektverwaltung für die Projektkontrolle und -steuerung zu übernehmen.

LPM06.S.3 Eingangs-Schnittstelle zu LPM05 „Kostenschätzung unterstützen“
Es ist möglich, mit LPM05 erfaßte Kostenschätzungen ohne weitere Transformation gezielt aus der Objektverwaltung für die Projektkontrolle und -steuerung zu übernehmen.

3.9.6.3.2 Anforderungen an die Methodenunterstützung

LPM06.M.1 Earned Value Verfahren (EVV)

LPM06.M.1.1 Earned Value Diagram
Es ist möglich, den Verlauf der Entwicklung des Geldwertes der erbrachten Leistung, der angefallenen Kosten und die geplanten Kosten in Form eines Earned Value Diagrams darzustellen.

LPM06.M.2 Trend-Analyse (TRDA)

LPM06.M.2.1 Trend-Kurven

LPM06.M.2.1.1 Aufwand
Es ist möglich, Trend-Kurven bezogen auf den Aufwand zu erstellen.

LPM06.M.2.1.2 Termin
Es ist möglich, Trend-Kurven bezogen auf den Termin zu erstellen.

3.9.6.3.3 Anforderungen an die Funktionen

LPM06.F.1 Auswertungen
Auswertungen sind auf wählbaren Ausschnitten aus der Datenbasis möglich (z. B. bezogen auf Projekt, Aktivität, Ressource).

LPM06.F.2 Berichte

LPM06.F.2.1 Layout
Der Berichtsgenerator erlaubt die Festlegung eines einheitlichen Seitenaufbaus.

LPM06.F.2.2 Struktur
Es ist möglich, Berichte bzgl. ihrer Inhalte frei zu definieren.

LPM06.F.2.3 Adressatenkreis
Es ist möglich, Berichte adressatengerecht aus der Datenbasis zu generieren.

LPM06.F.2.4 Berichtszeitraum
Die Wahl eines Zeitfensters ist möglich.

Ein Zeitfenster schränkt die Betrachtung auf einen Berichtszeitraum ein.

LPM06.F.3 Statistische Auswertungen

LPM06.F.3.1 Auswertungen pro Aktivität
Es ist möglich, statistische Auswertungen bezogen auf Aktivitäten durchzuführen.

LPM06.F.3.2 Auswertungen pro Ressource
Es ist möglich, statistische Auswertungen bezogen auf Ressourcen durchzuführen.

LPM06.F.3.3 Auswertungen pro Produkt
Es ist möglich, statistische Auswertungen bezogen auf Produkte durchzuführen.

LPM06.F.3.4 Auswertungen für das gesamte Projekt
Es ist möglich, statistische Auswertungen bezogen auf das gesamte Projekt durchzuführen.

LPM06.F.3.5 Auswertungen für beliebige Zeitfenster
Es ist möglich, statistische Auswertungen bezogen auf beliebige Zeitfenster durchzuführen.

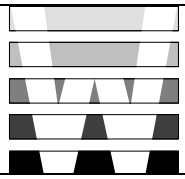
LPM06.F.4 Planvergleiche
Es ist möglich, verschiedene Planversionen zu vergleichen.

LPM06.F.5 Simulation
Es ist möglich, unterschiedliche Szenarien zu simulieren.

LPM06.F.6 Terminüberschreitungen
Es ist möglich, automatisch Terminüberschreitungen anzuzeigen.

3.9.6.3.4 Sonstige Anforderungen

LPM06.So.1 Erfahrungsdatenbank
Es ist möglich, Plan- und Ist-Werte in einer Erfahrungsdatenbank abzulegen.



LPM06.So.2 Aufwärtskompatibilität
Objekte, die mit einer früheren Version des Werkzeugs erstellt wurden, müssen mit dem neueren ohne Informations- und Funktionalitätsverlust verarbeitet werden können.

LPM06.So.3 Prozedurale Kommandosprache
Das Werkzeug besitzt eine prozedurale Kommandosprache, die der Nutzer verwenden kann, um Makros bzw. Prozeduren zu erstellen und ablaufen zu lassen.

LPM06.So.4 Komplexität
Eine Beschränkung der Komplexität ist durch das Werkzeug selbst nicht gegeben.